remote sensing

MDPI

Article

Unsupervised Feature-Learning for Hyperspectral Data with Autoencoders

Lloyd Windrim [1,*], Rishi Ramakrishnan [2,†], Arman Melkumyan [1] and Richard J. Murphy [1] and Anna Chlingaryan [1]

[1] Australian Centre for Field Robotics, University of Sydney, Sydney 2006, Australia; a.melkumyan@acfr.usyd.edu.au (A.M.); richard.murphy@sydney.edu.au (R.J.M.); a.chlingaryan@acfr.usyd.edu.au (A.C.)

[2] Baymatob Operations Pty Ltd, Leichhardt, Sydney 2040, Australia; rishi.ramakrishnanrr@gmail.com

* Correspondence: l.windrim@acfr.usyd.edu.au

† Work done while at Australian Centre for Field Robotics.

Received: 27 February 2019; Accepted: 3 April 2019; Published: 10 April 2019

Abstract: This paper proposes novel autoencoders for unsupervised feature-learning from hyperspectral data. Hyperspectral data typically have many dimensions and a significant amount of variability such that many data points are required to represent the distribution of the data. This poses challenges for higher-level algorithms which use the hyperspectral data (e.g., those that map the environment). Feature-learning mitigates this by projecting the data into a lower-dimensional space where the important information is either preserved or enhanced. In many applications, the amount of labelled hyperspectral data that can be acquired is limited. Hence, there is a need for feature-learning algorithms to be unsupervised. This work proposes unsupervised techniques that incorporate spectral measures from the remote-sensing literature into the objective functions of autoencoder feature learners. The proposed techniques are evaluated on the separability of their feature spaces as well as on their application as features for a clustering task, where they are compared against other unsupervised feature-learning approaches on several different datasets. The results show that autoencoders using spectral measures outperform those using the standard squared-error objective function for unsupervised hyperspectral feature-learning.

Keywords: autoencoders; unsupervised feature-learning; hyperspectral; deep learning

1. Introduction

Data acquired using a hyperspectral camera contains a plethora of information related to the chemical and physical properties of the in situ materials. Each pixel of a hyperspectral image has many dimensions pertaining to the absorption characteristics of a material at specific wavelengths in the visible, near infrared, and short-wave infrared (SWIR). As such, the data have been used for a variety of remote-sensing tasks, such as mapping/classification of surface materials [1–3], target detection [4,5] and change detection [6]. Because of the information content of each pixel in a hyperspectral image, the advantage of using hyperspectral imagery over conventional RGB and multispectral cameras is that many of the aforementioned tasks can be done at the pixel level.

The high dimensionality of hyperspectral data can also be problematic. With more dimensions, exponentially more data points are required to accurately represent the distribution of the data (i.e., the curse of dimensionality [7–9]). A low ratio between the number of data points and the number of dimensions may limit many algorithms from working well where the data have many dimensions. Most of the mass of a high dimensional multivariate Gaussian distribution is near its edges. Thus, many algorithms designed around an intuitive idea of 'distance' in two- or three-dimensional space, cease to work at higher dimensions, where those intuitions no longer hold. This problem is compounded by variability present in a hyperspectral image due to intrinsic factors such as keystone, smile, and noise and extrinsic factors such as the incident illumination. The incident illumination is dependent on the position of the sun, surface geometry and prevailing atmospheric conditions [10]. This variability demands more data points to adequately represent the distribution of the data.

It is often the case that key information necessary for a high-level task such as classification resides either in only a subset of the dimensions or in a lower-dimensional subspace within the high dimensional space. For example, proximal wavelengths in hyperspectral data are usually highly correlated, resulting in many redundant dimensions [11]. Features that are either selected or extracted (via some transformation) from the data in the spectral domain leverage this to represent the key information in the data with fewer dimensions. In many cases, a good feature space will actually enhance the important information, simplifying the task for higher-level algorithms using the data. For example, a good feature space for a classification task will separate distributions of points belonging to different classes (increasing inter-class variation), while making the distribution for each given class more compact in the space (decreasing intra-variation) [12]. In the case of hyperspectral data, the feature space should ideally have greater representation power than the raw intensity or reflectance space.

One approach for obtaining features from hyperspectral data is to manually hand-craft them. In [13], layers of clay on a mine face are mapped using the width and depth of a spectral absorption feature at 2200 nm. The Normalized Difference Vegetation Index (NDVI), which is a simple function of the reflectance at two wavelengths, is typically used for mapping vegetation [14]. While these features usually work very well for the application they were designed for, they are often difficult to design, require sufficient expertise and at times do not generalize well to new scenarios.

An alternative approach to obtaining features is to learn them directly from the data. This can be done using either supervised methods (e.g., Linear Discriminant Analysis (LDA) [15], kernel methods [16], autoencoder with logistic regression layer [17]) or unsupervised methods (e.g., Principal Component Analysis (PCA) [18,19], Independent Component Analysis (ICA) [20], basis functions [21]). Supervised feature-learning methods require labelled data. In many applications labelled hyperspectral data is limited due to the challenges associated with annotation [22]. This motivates a need for unsupervised feature-learning methods which extract important information from the data without the need for any labelling. Unsupervised learning methods are particularly good as a feature extractor for higher-level unsupervised tasks such as clustering. Clustering, as opposed to classification, can be used to map a scene without the need for any labelled data [23,24]. Thus, an entire pipeline, from the feature-learning to the mapping, can be unsupervised.

Autoencoders [25–27] are a useful tool for unsupervised feature-learning that have been used to obtain features from hyperspectral data [11,28–31]. An autoencoder is a neural network trained to reconstruct its input in the output. Through constraints imposed by the architecture, the network is forced to learn a condensed representation of the input that possess enough information to be able to reconstruct (i.e., decode) it again. This condensed layer can be used as a feature representation. With multiple layers of non-linear functions in the encoder and decoder, the network can learn a non-linear feature mapping.

When designing an autoencoder for hyperspectral data, a common approach, and the one used in [11,28–31], is to train the network to minimize a reconstruction objective based on a squared-error function. A squared-error function works generically well in most domains—not just hyperspectral. However, in the remote-sensing literature, there are alternative error measures more suited to spectral data. Unlike the squared-error measure, these measures are predominantly dependent on the shape of the spectra rather than their magnitude. Previous work [32] has incorporated a spectral measure, the cosine of the spectral angle, into the reconstruction objective of an autoencoder. The work presented in this paper further explores this space with novel autoencoders proposed which have spectral measures from the remote-sensing literature, the Spectral Information Divergence (SID) [33] and the spectral angle [34,35], incorporated into the reconstruction objective.

The contributions of this work are:

- Two novel autoencoders that use remote-sensing measures, the SID, and spectral angle, for unsupervised feature-learning from hyperspectral data in the spectral domain.
- An experimental comparison of these techniques with other unsupervised feature-learning techniques on a range of datasets.
- Further experimental evaluation of the autoencoder based on the cosine of the spectral angle proposed in previous work [32].

The outline of this paper is as follows: Section 2 provides a background on applying autoencoders to hyperspectral data, Section 3 describes the proposed autoencoders, Section 4 provides the experimental method and results which are discussed in Section 5 before conclusions are drawn in Section 6.

2. Background

An autoencoder is a special case of a Multi-layer Perceptron (MLP), which regresses the input data (or some variant of the input) in its output layer [26]. In doing so, it requires no labelled data, making it unsupervised, and can be used to learn non-linear features from the data. It comprises an encoder stage, a code layer, and a decoder stage. With one or more hidden layers, the encoder stage maps the input data to the code layer, which usually has fewer neurons than the input layer. The code must capture all the vital information needed to reconstruct the input in the output layer, via the decoder stage. The encoder and decoder stages are typically symmetric. The autoencoder, as with the MLP, comprises a network of trainable weights. The weights are optimized using a gradient descent process which aims to minimize an objective function, which usually includes a reconstruction error term and a regularization term. The partial derivatives of the objective function with respect to each of the parameters in the network necessary for gradient descent are computed using backpropagation. The network parameters are updated to minimize the objective score, typically until it converges. Hyperspectral data are input into an autoencoder as individual spectra, where the reflectance or intensity at each wavelength corresponds to the value of each neuron in the input layer (see Figure 1).

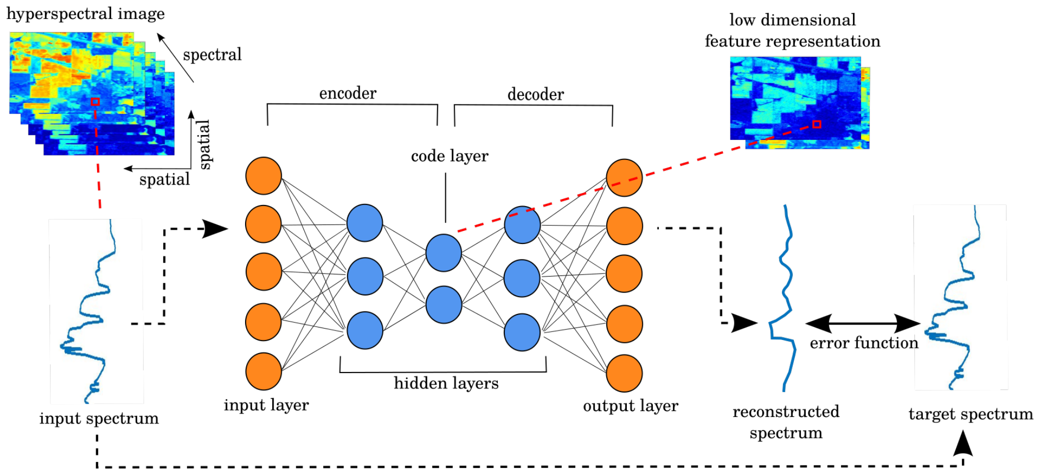


Figure 1. A simple example of how an autoencoder can be used for unsupervised learning of features from hyperspectral data. An individual spectrum constitutes a single training example which is both input into the network and used as a reconstruction target. The reflectance or intensity at each wavelength corresponds to the value of each input neuron (in this example there are only five input neurons, but in practice there would most likely be more). As the network learns to reconstruct the target, the code layer develops into a powerful yet condense feature representation of each spectrum (two neurons corresponds to a feature space of dimensionality two). Once trained, the encoder stage of the network can be used to map input spectra to the new feature space, which has fewer dimensions if the number of neurons in the code layer is fewer than the number in the input layer.

3. Novel Autoencoders for Unsupervised Feature-Learning

Two novel autoencoders are proposed for unsupervised feature-learning from hyperspectral data. These autoencoders use remote-sensing measures for their error functions (Figure 1), which alters the features learnt. The Spectral Information Divergence Stacked Autoencoder (SID-SAE) incorporates the SID into its objective function, while the Spectral Angle Stacked Autoencoder (SA-SAE) uses the spectral angle. The partial derivatives for each objective function have been derived.

3.1. Spectral Information Divergence Stacked Autoencoder

The SID is an information-theoretic measure which determines the probabilistic discrepancy between two spectra to calculate their similarity. Experiments have shown that it can preserve spectral properties and characterize spectral variability more effectively than the spectral angle [33].

The SID between two one-dimensional spectra **A** and **B**, each with N bands (i.e., channels), is given by:

$$\mathrm{SID}(\mathbf{A},\mathbf{B}) = \sum_{n=1}^{N} p_n log \frac{p_n}{q_n} + \sum_{n=1}^{N} q_n log \frac{q_n}{p_n}, \tag{1}$$

where the vectors **p** and **q** are the spectra **A** and **B** normalized by their respective sums:

$$\mathbf{p} = \frac{\mathbf{A}}{\sum_{t=1}^{T} A_t}, \tag{2}$$

$$\mathbf{q} = \frac{\mathbf{B}}{\sum_{t=1}^{T} B_t}, \tag{3}$$

where A_t and B_t are elements of the vectors **A** and **B**, corresponding to spectral values at band t, and T is the number of elements in either **A** or **B** (which, as with N, corresponds to the number of bands). To incorporate the SID into the autoencoder objective (i.e., cost function), Equation (1) is first simplified to:

$$\mathrm{SID}(\mathbf{A},\mathbf{B}) = \sum_{n=1}^{N} (p_n - q_n)(log(p_n) - log(q_n)). \tag{4}$$

Then, by making **A** the reconstructed spectrum output by the network and **B** the spectrum input into the network, the relevant terms are substituted into Equation (4):

$$E_{SID}(f(\mathbf{z}^{(L)}),\mathbf{y}) = \sum_{k=1}^{K} \left[\frac{f(z_k^{(L)})}{\sum_{d=1}^{K} f(z_d^{(L)})} - \frac{y_k}{\sum_{d=1}^{K} y_d} \right] ...$$

$$... \left[log f(z_k^{(L)}) - log \sum_{d=1}^{K} f(z_d^{(L)}) - log(y_k) + log \sum_{d=1}^{K} y_d \right], \tag{5}$$

where K is the original dimensionality, L is the index of the output layer, y_k is an element of the target data **y** which is set to equal the input data **x**, f is the activation function, $f(z_k^{(L)})$ is an element of the reconstructed input $f(\mathbf{z}^{(L)})$ and:

$$\mathbf{z}^{(l)} = \mathbf{W}^{(l-1)}\mathbf{a}^{(l-1)} + \mathbf{b}^{(l-1)}, \tag{6}$$

$$\mathbf{a}^{(l)} = f(\mathbf{z}^{(l)}), \tag{7}$$

$$\mathbf{a}^{(1)} = \mathbf{x} \tag{8}$$

for $l = L, L-1, L-2, L-3, ..., 2$, with learnable parameters **W** and **b**.

The reconstruction cost function for all observations, including a regularization term is:

$$E(\mathbf{W},\mathbf{b}) = \frac{1}{M}\sum_{m=1}^{M} E_{SID}(\mathbf{W},\mathbf{b};\mathbf{y}^{(m)}) + \frac{\lambda}{2}\sum_{i,j,l=1}^{I,J,L-1}(W_{ji}^{(l)})^2. \tag{9}$$

where M is the number of observations, λ is the regularization parameter, and I and J are the number of units in layers l and $l+1$ respectively. The regularization term prevents the parameters from getting too large whereby over-fitting occurs. The partial derivatives for backpropagation are:

$$\frac{\partial}{\partial W_{ji}^{(l)}}E(\mathbf{W},\mathbf{b}) = \frac{1}{M}\sum_{m=1}^{M}\frac{\partial}{\partial W_{ji}^{(l)}}E_{SID}(\mathbf{W},\mathbf{b};\mathbf{y}^{(m)}) + \lambda W_{ji}^{(l)}, \tag{10}$$

$$\frac{\partial}{\partial b_{j}^{(l)}}E(\mathbf{W},\mathbf{b}) = \frac{1}{M}\sum_{m=1}^{M}\frac{\partial}{\partial b_{j}^{(l)}}E_{SID}(\mathbf{W},\mathbf{b};\mathbf{y}^{(m)}), \tag{11}$$

where for a single observation m:

$$\frac{\partial}{\partial W_{ji}^{(l)}}E_{SID}(\mathbf{W},\mathbf{b};\mathbf{y}^{(m)}) = \delta_j^{(l+1)}a_i^{(l)}, \tag{12}$$

$$\frac{\partial}{\partial b_{j}^{(l)}}E_{SID}(\mathbf{W},\mathbf{b};\mathbf{y}^{(m)}) = \delta_j^{(l+1)}, \tag{13}$$

for $l = 1,2,3,...,L-1$. The value of δ is dependent on the layer number l. For $l = L$,

$$\delta_k^{(L)} = -\frac{f'(z_k^{(L)})}{\sum_{d=1}^{K} f(z_d^{(L)})}\left[\frac{q_k}{p_k} - log\frac{p_k}{q_k} - 1 + \sum_{d=1}^{K}(p_d - q_d + p_d log\frac{p_d}{q_d})\right], \tag{14}$$

where

$$\mathbf{p} = \frac{f(\mathbf{z}^{(L)})}{\sum_{c=1}^{K} f(z_c^{(L)})}, \tag{15}$$

$$\mathbf{q} = \frac{\mathbf{y}}{\sum_{c=1}^{K} y_c}, \tag{16}$$

and for $l = L-1, L-2, L-3, ..., 2$,

$$\delta_i^{(l)} = \sum_{j=1}^{J}(\delta_j^{(l+1)}W_{ji}^{(l)})f'(z_i^{(l)}). \tag{17}$$

The parameter update equations for gradient descent optimization are:

$$W_{ji}^{(l)} := W_{ji}^{(l)} - \alpha\frac{\partial}{\partial W_{ji}^{(l)}}E(\mathbf{W},\mathbf{b}), \tag{18}$$

$$b_{j}^{(l)} := b_{j}^{(l)} - \alpha\frac{\partial}{\partial b_{j}^{(l)}}E(\mathbf{W},\mathbf{b}), \tag{19}$$

where α is the learning rate. This approach is called the SID-SAE.

3.2. Spectral Angle Stacked Autoencoder

The spectral angle [34,35] is a similarity measure between two vectors. The angle between vectors is indicative of their shape rather than their magnitude, making it a useful measure for applications using spectral data. This measure is also insensitive to differences in brightness, which can be interpreted as changes in the magnitude of the spectral vectors, rather than changes in the direction of the vectors in multi-dimensional space. Hence, the angle remains the same. By using a reconstruction cost function for an autoencoder based on the spectral angle instead of the squared error, the features learnt are more sensitive to the shape of the spectral curve and insensitive to variations in the brightness of spectra [36].

The spectral angle θ_{SA} is the angular distance between two spectral vectors, $\mathbf{A}$ and $\mathbf{B}$, each with T bands, given by:

$$\theta_{SA} = \cos^{-1} \frac{\sum_{t=1}^{T} A_t B_t}{|\mathbf{A}||\mathbf{B}|}. \tag{20}$$

In previous work [32], the cosine of the spectral angle was incorporated into an autoencoder:

$$\cos(\theta_{SA}) = \frac{\sum_{t=1}^{T} A_t B_t}{|\mathbf{A}||\mathbf{B}|}. \tag{21}$$

In this work, we incorporate the actual angle (i.e., Equation (20)) rather than its cosine, and compare its performance. All further references to the autoencoder published in [32] will be as the Cosine Spectral Angle Stacked Autoencoder (CSA-SAE).

The spectral angle is incorporated into the reconstruction cost function by calculating the spectral angle between the reconstructed spectrum output by the encoder-decoder network and the spectrum input into the network. The autoencoder is trained by minimizing the overall cost. For a single observation, the reconstruction cost is:

$$E_{SA}(f(\mathbf{z}^{(L)}), \mathbf{y}) = \cos^{-1} \frac{\sum_{k=1}^{K} f(z_k^{(L)}) y_k}{|f(\mathbf{z}^{(L)})||\mathbf{y}|}, \tag{22}$$

and the reconstruction cost function for all observations, including a regularization term is:

$$E(\mathbf{W}, \mathbf{b}) = \frac{1}{M} \sum_{m=1}^{M} E_{SA}(\mathbf{W}, \mathbf{b}; \mathbf{y}^{(m)}) + \frac{\lambda}{2} \sum_{i,j,l=1}^{I,J,L-1} (W_{ji}^{(l)})^2. \tag{23}$$

As with the autoencoder based on the SID, the partial derivatives for backpropagation are the same as in Equations (10) and (11), but with E_{SA} instead of E_{SID} and δ for $l = L$ calculated differently:

$$\delta_k^{(L)} = \frac{1}{\sqrt{1 - \left[\frac{(f(\mathbf{z}^{(L)}) \cdot \mathbf{y})}{|f(\mathbf{z}^{(L)})||\mathbf{y}|}\right]^2}} \frac{f'(z_k^{(L)})}{|f(\mathbf{z}^{(L)})||\mathbf{y}|} \left[\frac{(f(\mathbf{z}^{(L)}) \cdot \mathbf{y}) f(z_k^{(L)})}{|f(\mathbf{z}^{(L)})|^2} - y_k \right], \tag{24}$$

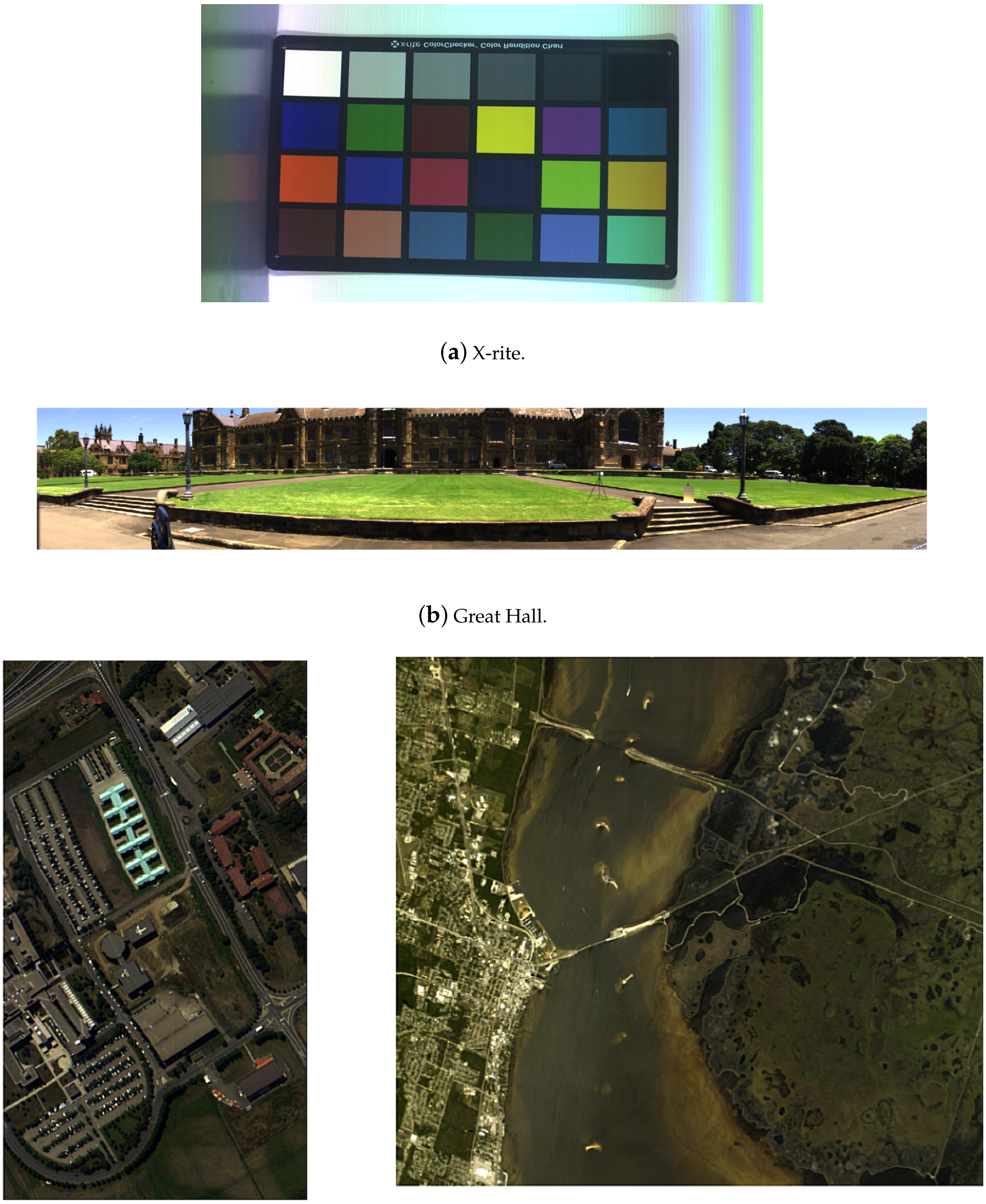
with Equation (17) still applying for $l = L-1, L-2, L-3, ..., 2$. The parameter update equations are once again as in (18) and (19). This approach is called the SA-SAE

4. Experimental Results

The performance of the features learnt with the proposed hyperspectral autoencoders were evaluated experimentally with several datasets to demonstrate the algorithm's ability to work with different scenes, illumination conditions and sensors used to capture the images. The evaluation involved an analysis of the separability of the feature spaces as well as the performance of the feature spaces for clustering and classification tasks.

4.1. Datasets

The datasets used to evaluate the proposed feature learners were captured from different distances to the scene and hence have different spatial resolutions. They were also captured from both airborne and field-based sensor platforms. The scenes covered simple, structured and complex, unstructured geometries, and exemplified a variety of illumination conditions. Table 1 summarizes the datasets used, and Figure 2 visualizes them as color composites. For these datasets, each multi-channel pixel of a hyperspectral image corresponds to a data point (i.e., a single spectrum).



(**a**) X-rite.

(**b**) Great Hall.

(**c**) Pavia University.

(**d**) KSC.

Figure 2. Color composite images of the hyperspectral datasets used for the experiments.

Table 1. A summary of the datasets used in the experiments. Datasets were captured with different cameras and with different scene contents to extensively evaluate the generality of the proposed algorithms.

Dataset Name	Spectrum	Sensor	Spectral Range (nm)	No. channels	Spectral res. (nm)	No. Classes	No. pixels
Simulated USGS	VNIR + SWIR	-	383–2496	424	2–10	10	$1 \times 10,000$
X-rite [37]	VNIR	Specim	378–1003	396	2	24	697×1312
Great Hall [38]	VNIR	Specim	400–1007	132	4	6	320×2010
Pavia University	VNIR	ROSIS-3	430–860	103	6	9	610×340
KSC	VNIR + SWIR	AVIRIS	400–2500	176	10	13	512×614

Mineral spectra from the USGS spectral library [39] were used to simulate a dataset. Ten minerals were selected to generate spectral samples (Figure 3). The data was simulated using the principle that the radiance at the sensor $\mathbf{L}$ is the product of the material reflectance $\boldsymbol{\rho}$ and the terrestrial sunlight $E_{sun}(\lambda)\tau(\lambda)$ with some additive noise:

$$L(\lambda) = I(\lambda)\rho(\lambda)E_{sun}(\lambda)\tau(\lambda) + noise, \tag{25}$$

where $\mathbf{I}$ is a scaling factor controlling the brightness. The terrestrial sunlight was generated using an atmospheric modeler [40]. Spectra from a calibration panel were also simulated. These data are used to normalize the simulated radiance data to reflectance. Each dataset consisted of 10,000 spectral samples (1000 samples for each of the ten classes in Figure 3).

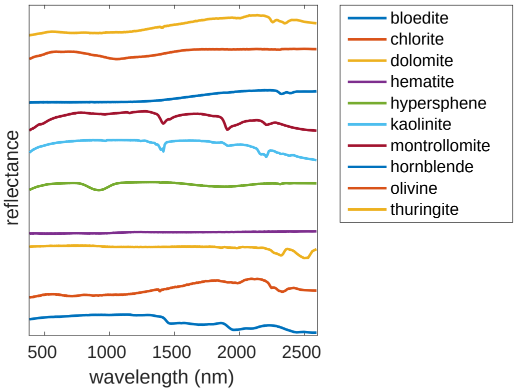


Figure 3. Simulated USGS dataset with ten different mineral classes. The spectra are offset from each other in the vertical axis for clarity. Reflectance values range from 0 to 1.

The X-rite ColorChecker is an array of 24 different colored squares. They reflect light in the visible spectrum in the same way as many naturally colored objects. A Visible and Near Infrared (VNIR) image of an X-rite colorChecker was captured indoors with a Specim sensor under a 3500 K temperature light source [37]. The geometry is very uniform and there is little cause for variation in the illumination. The image was normalized to reflectance using the white square on the colorChecker.

A VNIR scan of The University of Sydney's Great Hall was captured with a field-based Specim AISA Eagle sensor [38]. The outdoor scene consists predominantly of a structured urban environment, with different material classes including roof, sandstone building, grass, path, tree, and sky. The roof and path classes are very similar spectrally; however, the other classes are quite discriminable, including the tree and grass. Data were collected under clear-sky conditions, with shadows being evenly distributed across the structure. A calibration panel of known reflectance is attached to a tripod and placed in the scene so that the incident illumination can be measured. The apparent reflectance across the image was calculated using flat-field correction.

An aerial VNIR image is acquired over Pavia University using the ROSIS-3 sensor. The spectra in the image have been normalized to reflectance. The scene consists of nine classes, encompassing both urban and natural vegetation classes. Some of the classes are spectrally similar. The Pavia University dataset has been provided by Professor Paolo Gamba, Pavia University.

The Kennedy Space Centre (KSC) image was acquired by the AVIRIS VNIR/SWIR sensor. The water absorption bands have been removed from this image. The 13 classes are a mix of urban and natural and the data are normalized to reflectance.

4.2. Evaluation Metrics

Three metrics were used to evaluate the performance of the features extracted from the above datasets. These were the Fisher's discriminant ratio, Adjusted Rand Index (ARI) and F1 score.

Fisher's discriminant ratio is a measure of how separable classes are in a feature space. Fisher's discriminant ratio [41] is used in feature selection algorithms as a class separability criterion [42,43]. The measure is calculated for a pair of classes and takes the ratio of the between-class scatter and the within-class scatter. A pair of classes has a high score if their means are far apart and points within each class are close to other points in the same class, indicating good separability. For p dimensional data points from class A and class B, with respective means of $\boldsymbol{\mu_A}$ and $\boldsymbol{\mu_B}$ over all points in each class, the Fisher's discriminant ratio is calculated as:

$$J(A,B) = \frac{\|\boldsymbol{\mu_A} - \boldsymbol{\mu_B}\|_2^2}{S_A^2 + S_B^2}, \tag{26}$$

where J is the ratio, $\|\cdot\|_2$ is the L_2 norm, and S_i^2 is the within-class scatter of class i, given by:

$$S_i^2 = \frac{1}{N_i} \sum_{n \in N_i} \|\mathbf{x_n} - \boldsymbol{\mu_i}\|_2^2, \tag{27}$$

where $\mathbf{x_n}$ is a point in class i, which has N_i points in total.

An important property of this measure is that it is invariant to the scale of the data points. This allows feature spaces found using different approaches to be compared consistently. It is also important that the measure is invariant to the number of dimensions p so that the data with the original dimensionality can be compared to data with reduced dimensionality in the new feature space. For multi-class problems, the mean of the Fisher's discriminant ratio for all possible pairs of classes can be found.

The ARI is an evaluation metric for clustering [44,45]. It measures the similarity of two data clusterings. If one of those data clustering is the actual class labels, then the ARI can be considered the accuracy of the clustering solution. This measure is useful as an indirect measure of how separable classes are in a feature space [46]. If the classes are well separated, then they should be easy to cluster, and the ARI should be high.

If n_i and n_j are the number of points in class u_i and cluster v_j respectively and n_{ij} is the number of points that are in both class u_i and cluster v_j, then the ARI is calculated as:

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{n_i}{2} \sum_j \binom{n_j}{2}\right] / \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{n_i}{2} + \sum_j \binom{n_j}{2}\right] - \left[\sum_i \binom{n_i}{2} \sum_j \binom{n_j}{2}\right] / \binom{n}{2}}. \tag{28}$$

This adjustment of the rand index makes it more likely that random label assignments will get a value close to zero. The best clustering score has an ARI of 1, and the worst clustering score (equivalent to randomly allocating cluster association) has an adjusted rand index of 0.

The F1 score (also known as the F-score) is a commonly use evaluation metric for classification accuracy [47]. It is calculated as the harmonic mean of precision and recall:

$$F_1score = 2 \times \frac{precision \times recall}{precision + recall}, \tag{29}$$

where the precision and recall are defined as:

$$precision = \frac{TP}{TP + FP}, \tag{30}$$

$$recall = \frac{TP}{TP + FN}, \tag{31}$$

where TP, FP, and FN are the number of true positives, false positives, and false negatives respectively. The F1 score ranges from 0 to 1, with 1 indicating perfect classification performance. The F1 score is calculated separately for each class.

4.3. Implementation

The network architecture and tunable parameters of the autoencoder were found in preliminary experiments using a coarse grid search, minimizing the reconstruction error. A common architecture was chosen for the experiments that performed well across all the datasets. For the Great Hall VNIR, KSC, Pavia Uni, Simulated and X-rite datasets, unless otherwise stated, an encoder architecture of K-100-50-10 was used with symmetric decoders (six layers in total), where the input size K is the original dimensionality of the data (i.e., number of bands). Because the code layer has 10 neurons, the dimensionality of the data in the new feature space is 10.

For all datasets, the regularization parameter was set to 10^{-4}, the activation function used was a sigmoid and 1000 epochs of L-BFGS [48] was used to optimize the networks. As this method is unsupervised, the method is trained on the same image it is evaluated on (similar to the evaluation of a dimensionality-reduction approach). Hence, for each experiment, the number of data points used to train the autoencoders was the number of pixels in the image.

As is the case with standard SAEs, training a network with many layers using backpropagation often results in poor solutions [25]. Hence, an initial solution is found using a greedy pre-training step [49] whereby layers are trained in turn while keeping other layers frozen. The network parameters learnt in the layerwise pre-training are then used to initialize the full network for end-to-end fine-tuning with the same data, whereby all layers are trained at the same time. In [50], it was showed that it is possible to also pre-train networks on different hyperspectral datasets.

A property of the spectral angle reconstruction cost is that it is undefined for $|f(\mathbf{z}^{(L)})| = 0$. Hence, an activation function f must be chosen that does not include zero in its range. This removes the chance of having an undefined reconstruction cost. A sigmoid function is chosen as the activation function because it is bound by a horizontal asymptote at zero such that its range is $0 < f(x) < 1$. It is, therefore, impossible for $|f(\mathbf{z}^{(L)})|$ to equal zero, which is not the case if functions such as ReLU and the hyperbolic tangent are used.

All experiments were carried out in MATLAB.

4.4. Results

The proposed autoencoders, the SID-SAE and SA-SAE, were compared with other unsupervised feature-extraction/dimensionality-reduction methods based on their ability to discriminate spectra of different classes and similarly represent spectra of the same class with a reduced number of dimensions. These included PCA, Factor Analysis (FA), an equivalent autoencoder (referred to as SSE-SAE) that used the squared error for its reconstruction cost function (similar to [11,28–31]), an autoencoder using the cosine of the spectral angle as in [32] (referred to as CSA-SAE) and the raw data without any dimensionality reduction. The degree of variability in the scene differed between datasets, and the algorithms were evaluated on how robust they were to this variability. All the chosen datasets were used in normalized reflectance form, and the simulated and Great Hall VNIR datasets were also evaluated in Digital Number (DN) form.

The first set of results compared each methods ability to represent different classes with fewer dimensions. It is desirable that in the new feature space, spectra from different classes are separated and spectra from the same class are closer together. This was measured with the Fisher's discriminant ratio. If classes are well represented in the low-dimensional space, then it is expected that the data will cluster into semantically meaningful groups (rather than groups that have a similar incident illumination). Hence, as an additional method of evaluation, the low-dimensional data was clustered using k-means [51], where the number of clusters was set to the number of classes in the dataset. The clustering performance was measured using the ARI. Furthermore, the classification accuracy when using the unsupervised features with several common supervised classifiers was used to evaluate the proposed methods. The classifiers used were Support Vector Machines (SVM), k-Nearest Neighbors (KNN) and Decision Trees (DT). To train the classifier, the training sets comprised 1000, 500, and 50 data points, the validation sets comprised 100, 50, and 5 and the test sets comprised the remainder of the points for the Great Hall, Pavia Uni and KSC datasets respectively (these values were chosen based on the class with the fewest number of labelled points for each dataset). The classification accuracy was measured using the F1 score.

The Fisher's ratio results showed that for most datasets, the hyperspectral autoencoders that utilized remote-sensing measures (SID-SAE, SA-SAE and CSA-SAE) represented the different classes with fewer dimensions with more between-class discriminability and within-class uniformity than the other approaches (Figure 4). Clustering performance was also higher when using these autoencoders to represent the data (Figure 5). The hyperspectral autoencoder based on the cosine of the spectral angle (CSA-SAE) had the best overall performance. The results of Figure 4 indicated that this method performed well in comparison to the other methods on the simulated reflectance, simulated DN, Great Hall VNIR reflectance, Great Hall VNIR DN, and Pavia Uni datasets. The clustering results of Figure 5 supported the Fisher's ratio results in terms of relative performance, especially in comparison to the no-dimensionality reduction, PCA and FA results. The CSA-SAE had ARI scores of above 0.7 for the simulated reflectance dataset, Great Hall VNIR reflectance and Great Hall VNIR DN datasets. In terms of Fisher's score, the SID-SAE approach performed well on the simulated reflectance, simulated DN and Pavia Uni datasets, but only performed marginally better than the standard autoencoder, the SSE-SAE, on the other datasets. All methods had similar class separation on the Pavia Uni and simulated DN datasets, and the clustering performance of all methods for these datasets was low (ARI below 0.4). FA had the best separation for the KSC and X-rite 3500K datasets. There appeared to be no real advantage to using the raw spectral angle (SA-SAE) over the cosine of the spectral angle (CSA-SAE).

For the classification results (Figure 6), for any given classifier there were no clear trends across the datasets as to which feature method produced the best result. Observations can be made, such as for DTs, the SA-SAE performs comparatively well across all three datasets. However, in the Pavia Uni dataset it is outperformed by PCA, and in the KSC dataset, it performs similarly to SSE-SAE and the original reflectance features (raw spectra). In other cases, such as applying the SVM to the KSC dataset, the original reflectance features performed well (in this case it outperformed all other methods by a large margin). For the Great Hall dataset, all methods performed similarly well except for the CSA-SAE, which produced results that were slightly worse than the other methods.

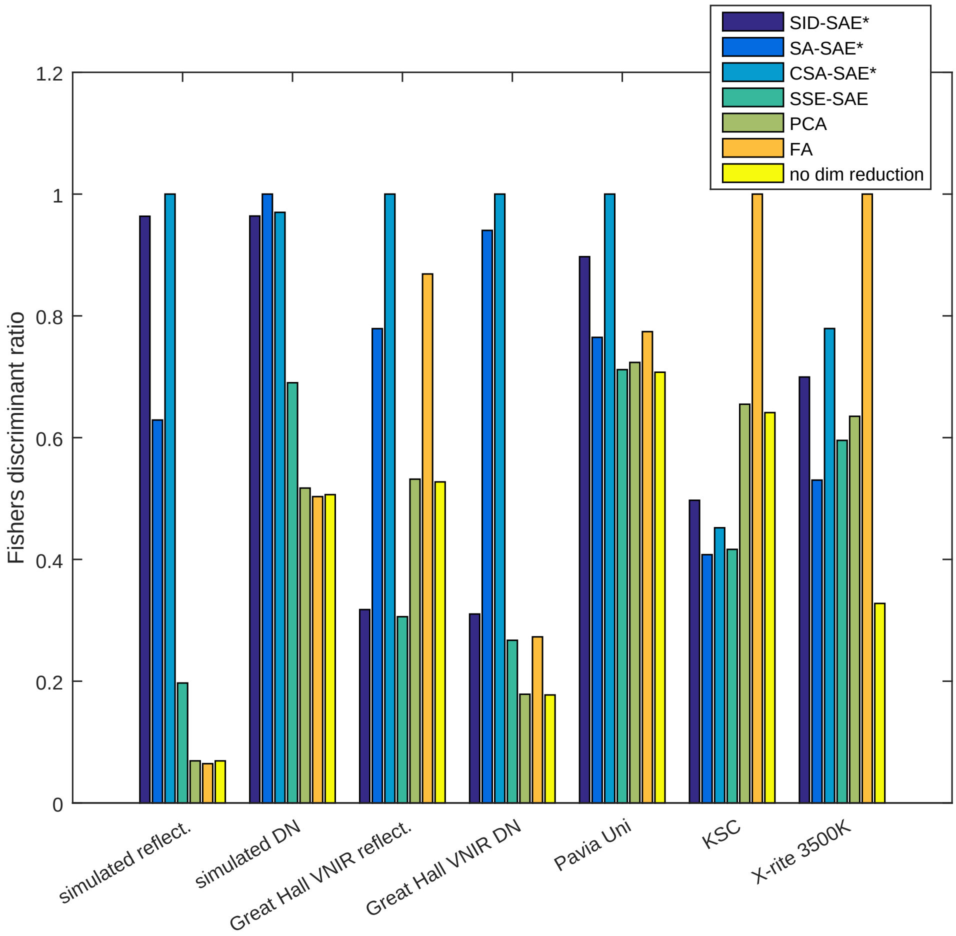


Figure 4. Comparison of representation power of different unsupervised feature-extraction/ dimensionality-reduction methods, for a range of different datasets (horizontal axis). The methods were evaluated on how well different classes were represented in the low-dimensional space using the Fisher's discriminant ratio. The higher the score, the better the representation because spectra from different classes were more separated relative to spectra from the same class. Scores have been standardized for comparison across datasets by dividing each score by the maximum achieved for that dataset. The * in the legend indicates the autoencoder methods using remote-sensing measures (of which the SID-SAE and SA-SAE were proposed in this paper).

Figure 7 visualizes how invariant the low-dimensional feature representations were to changes in brightness. Using the data that was simulated with different brightness, dimensionality-reduction mappings were learnt by training on data illuminated by a source with a given intensity (scaling factor equal to 1), and then applied to two different datasets: the said dataset as well as the same dataset illuminated by a source with a different intensity (scaling factor equal to 0.3). Figure 7 compares the low-dimensional representation of four different material spectra under the two different illumination intensities, for each of the autoencoder approaches. Bright refers to the dataset with scaling factor equal to 1, and dark has a scaling factor equal to 0.3. If a feature is invariant, a material illuminated with different intensities should appear similar in the feature representation. In a real-world outdoor dataset, differences in brightness can arise due to the variations in geometry with respect to the sun, and also changes in the brightness of the sun (for example, in a dataset of the same scene captured at different times).

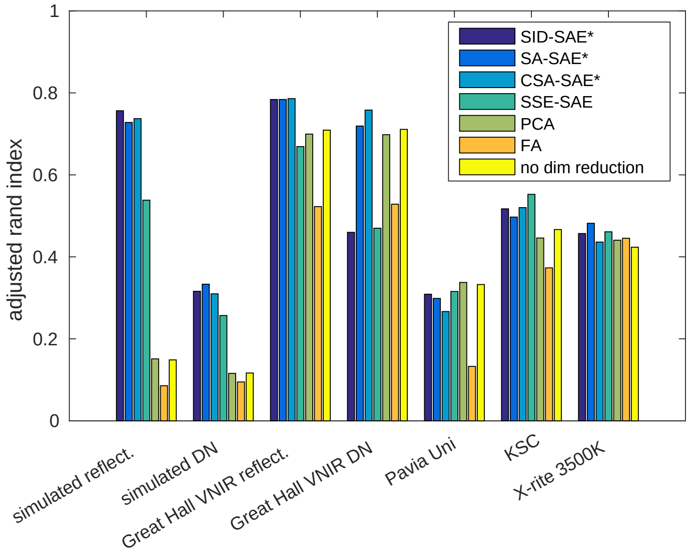


Figure 5. Comparison of clustering results after using different unsupervised feature-extraction/dimensionality-reduction methods, for a range of different datasets (horizontal axis). The methods were evaluated using the adjusted rand index of the clustered low-dimensional data. The higher the score, the better the clustering performance was. The * in the legend indicates the autoencoder methods using remote-sensing measures (of which the SID-SAE and SA-SAE were proposed in this paper).

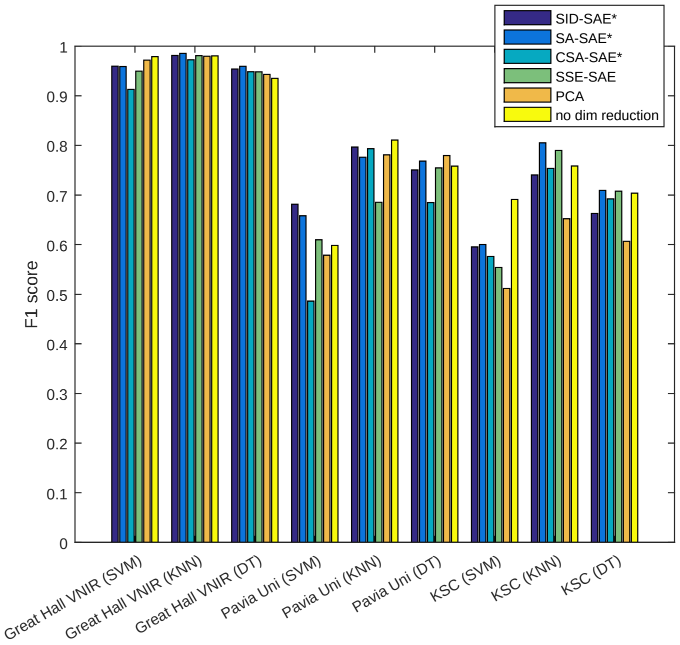
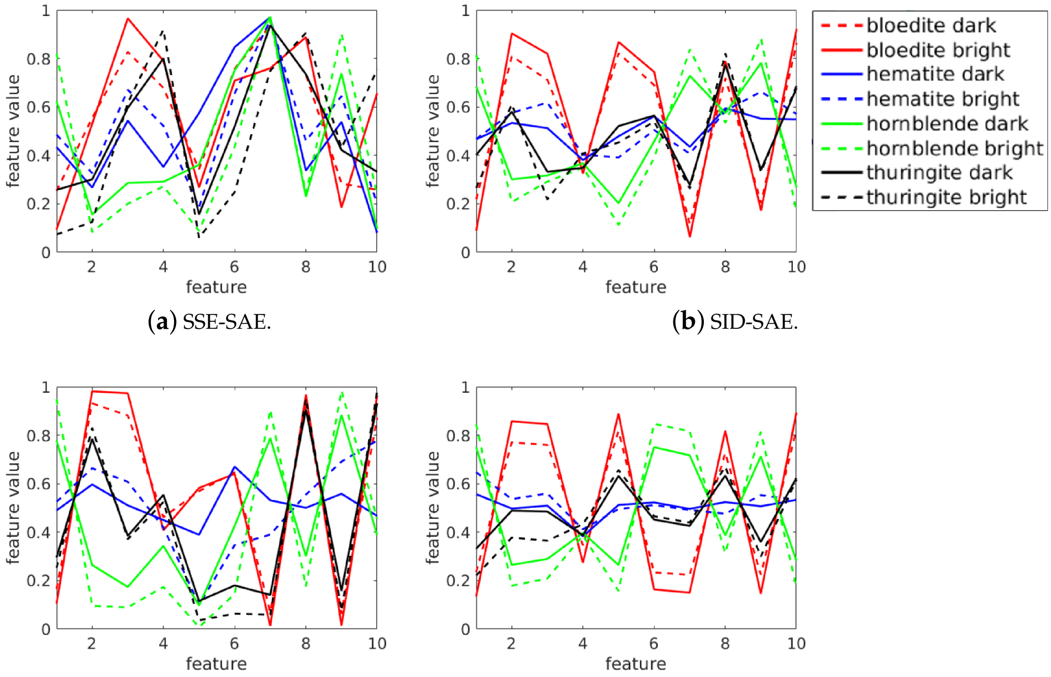


Figure 6. Comparison of classification results after using different unsupervised methods as features, for a range of different datasets and classifiers (horizontal axis). The classifiers used are Support Vector Machines (SVM), *k*-Nearest Neighbors (KNN) and Decision Trees (DT). The methods were evaluated using the mean F1 score over all classes. The higher the score, the better the classification performance was. The * in the legend indicates the autoencoder methods using remote-sensing measures (of which the SID-SAE and SA-SAE were proposed in this paper).



(**a**) SSE-SAE.

(**b**) SID-SAE.

(**c**) SA-SAE.

(**d**) CSA-SAE.

Figure 7. Comparison of low-dimensional autoencoder representation of different classes under different brightnesses, for the simulated reflectance dataset. Results show only four out of the ten classes.

Table 2 compares the performance of the learnt features using the two proposed hyperspectral autoencoder methods with different numbers of neurons in the code layer. Experiments were conducted on the Pavia University dataset, and code layer neurons varied from 6 to 14 with increments of 2 (the rest of the architecture remained the same). The results show that for a given method, there is not much variance when the number of nodes is varied. For the SID-SAE, the best score is achieved with 6 nodes, and for the SA-SAE, the best score is achieved with 14 nodes.

Table 2. Comparison of representation power of proposed hyperspectral autoencoders on the Pavia University dataset with different numbers of neurons in the code layer, evaluated with Fisher's discriminant ratio (used in the Figure 4 result). The higher the score, the better the representation. Scores have not been standardized in this case.

	Number of Neurons in Code Layer				
Method	**6**	**8**	**10**	**12**	**14**
SID-SAE	13.39	12.90	12.75	12.71	12.10
SA-SAE	9.98	9.49	10.04	10.67	10.80

5. Discussion

The autoencoder approaches were expected to have better overall performance than PCA as they were able to learn non-linear mappings to the low-dimensional spaces. This property allowed for more complex transformations to occur. The results (Figures 4 and 5) supported this for some of the datasets. The hyperspectral autoencoders that used remote-sensing measures had a better overall performance than the SSE-SAE, and further, the autoencoders based on the spectral angle performed better than the SID-SAE. By using the spectral angle as the reconstruction cost function, the autoencoders learnt complex mappings that captured features describing the shape of the spectra. This was in comparison to the autoencoder that used the squared-error measurement of reconstruction error. These autoencoders learnt features which described the intensity of the spectra. The intensity is highly dependent on the illumination conditions and hence was not the best characteristic on which features should be based on. The autoencoder using the SID performed slightly better than the SSE-SAE approach because the distance function could account for small variations in the probability distribution of each spectra and was also unaffected by wavelength-constant multiplications. Hence, the learnt mapping was invariant to intensity changes. FA performed best on the X-rite 3500K dataset. However, this was likely to be because the X-rite 3500K dataset was the only dataset that had almost no intraclass variability. Thus, features were learnt which maximized the variability between classes and there was very small separation created within each class.

In some cases, the hyperspectral autoencoders using remote-sensing methods did not outperform the other approaches. There was almost no performance improvement in Figure 4 from using the angle-based methods for the KSC and X-rite 3500 K datasets. However, the clustering results (Figure 5) were relatively low for these datasets for all methods, suggesting that they were particularly difficult to represent with unsupervised approaches. Clustering results were also low for the simulated DN and Pavia Uni datasets. The KSC, Pavia Uni and X-rite 3500 K datasets contain many classes (greater than eight), many of which are very similar. For example, the KSC dataset contains several different ′marsh′ classes. With that said, the hyperspectral autoencoders still outperformed the PCA, FA and no-dimensionality reduction approaches for the KSC dataset. Thus, the limitations of the proposed hyperspectral autoencoder are similar to those of any unsupervised method. In certain cases, such as when there are many similar classes as in the KSC dataset, a supervised method is more suitable. However, in situations where unsupervised learning techniques have utility, it is beneficial to use the proposed hyperspectral autoencoders.

Regarding the simulated DN dataset, the spectral angle methods were expected to achieve similar results on both the reflectance and DN datasets, because the shape of the spectra is unique for different classes regardless of normalization. For the Fisher's score results (Figure 4) this was true, but for the clustering results, this was only the case for the Great Hall datasets and not the simulated datasets. The clustering results show that all methods performed badly on the simulated DN dataset, suggesting that if not normalized, the differences in the spectra can become very small, making it difficult for any unsupervised methods to separate the classes.

From the results of Figure 6, there was no obvious feature and classifier combination that performed definitively better than other combinations across all datasets. Thus, it is difficult to draw conclusions about the benefit of using the proposed unsupervised feature methods with supervised classifiers. The added complexity of the classifier makes it hard to determine whether a combination worked well or poorly with a particular dataset due to the features or the classifier. From the results, in general the proposed features can be used with a classifier for adequate performance, but in many cases the raw spectra were equally as or more effective than using the features. It is possible that in some cases the hyperspectral features suppress information that is valuable for classification. For example, differences in the magnitude of the spectra could be due to brightness variations, promoting undesirable intraclass variation, but they could also indicate a different semantic class, such as the roof and path classes of the Great Hall dataset. These two classes have spectra with a similar shape, but with different magnitudes. Because the hyperspectral autoencoders emphasize spectral shape, they could be less effective at distinguishing these classes, where shape is not a valuable cue. This could explain the slightly worse performance of the CSA-SAE on the Great Hall dataset when using the SVM. The raw spectra perform well because it captures the differences in magnitude needed to distinguish the classes, but not at the cost of performance due to brightness variability because there are enough labelled training examples of each class under varying illumination for the supervised classifier to learn to generalize. Thus, it is recommended that if sufficient labelled data is available, it is best to use either the raw spectra or a supervised feature learner such as a neural network optimized with a classification loss function [22], rather than the unsupervised remote-sensing specific reconstruction loss functions proposed in this paper.

For the autoencoders using remote-sensing measures, the encodings of spectra with different brightnesses appeared similar in the learnt feature space (Figure 7), indicating a degree of invariance to brightness. Mappings learnt on simulated data, illuminated with a specific brightness, were able to generalize to simulated data illuminated with a brightness that the mappings were not trained on. This was expected from the proposed hyperspectral autoencoders because their reconstruction cost functions are robust to brightness variations. Interestingly, the SSE-SAE also exhibited some brightness invariance despite the reconstruction cost function being dependent on the intensity of the spectra. It was expected that this occurred because there was no brightness variability within the training set. As a result, the SSE-SAE learnt to ignore brightness variability because it did not require reconstruction. If, however, there was variability in the brightness of the training data then it is likely that the SSE-SAE would capture the very basic intensity multiplication in the training dataset with its non-linear function approximators and encode these variations in several features/dimensions. If the trained encoder was then applied to new data with a different intensity of the source brightness, then some of the features in the encoding would vary depending on the brightness.

The results of Table 2 indicate that there is flexibility in the choice of the number of nodes in the code layer, which is good for unsupervised tasks. This is because cues for the design of the code layer, such as the intrinsic dimensionality of the data or the number of classes, are often unknown. The results show a slight trend for both methods: that for the Pavia University dataset, the optimal number of neurons in the code layer is fewer for the SID-SAE than the SA-SAE. This could suggest that for this particular dataset, the SID-SAE is more efficient at representing the data. However, the performance margin (across neuron numbers) is too small to be able to conclude this decisively from the results.

6. Conclusions

This work proposed two novel methods for unsupervised feature-learning from hyperspectral data in the spectral domain. These methods incorporated two well studied measures of spectral similarity, the SID and spectral angle, from the remote-sensing literature into an autoencoder learning framework. Experiments to evaluate the proposed features showed that in general, autoencoders that use remote-sensing measures learn more discriminative feature representations for hyperspectral data than other unsupervised feature-learning methods, such as standard autoencoders that use a squared-error loss. They also showed some invariance to varying brightness. Techniques such as PCA simply reduce the number of dimensions while preserving the representation power of the feature space, whereas the autoencoders typically learn a much better feature space (shown to work well for an unsupervised clustering task).

Author Contributions: L.W. was the principal investigator and was responsible for designing the algorithms and experiments, analyzing the data and preparing the manuscript. R.R., A.M., R.J.M. and A.C. provided input into aspects of the algorithmic and experimental design.

Acknowledgments: This work has been supported by the Rio Tinto Centre for Mine Automation and the Australian Centre for Field Robotics, University of Sydney.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CSA-SAE	Cosine Spectral Angle Stacked Autoencoder
FA	Factor Analysis
KSC	Kennedy Space Centre
MLP	Multi-layer Perceptron
PCA	Principle Component Analysis
SA-SAE	Spectral Angle Stacked Autoencoder

SID	Spectral Information Divergence
SID-SAE	Spectral Information Divergence Stacked Autoencoder
SSE-SAE	Sum of Squared Errors Stacked Autoencoder
SWIR	Short-Wave Infrared
VNIR	Visible and Near Infrared

References

1. Murphy, R.J.; Monteiro, S.T.; Schneider, S. Evaluating Classification Techniques for Mapping Vertical Geology Using Field-Based Hyperspectral Sensors. *IEEE Trans. Geosci. Remote Sens.* **2012**, *50*, 3066–3080. [CrossRef]
2. Marcus, W.A.; Legleiter, C.J.; Aspinall, R.J.; Boardman, J.W.; Crabtree, R.L. High spatial resolution hyperspectral mapping of in-stream habitats, depths, and woody debris in mountain streams. *Geomorphology* **2003**, *55*, 363–380. [CrossRef]
3. Wendel, A.; Underwood, J. Self-Supervised Weed Detection in Vegetable Crops Using Ground Based Hyperspectral Imaging. In Proceedings of the IEEE International Conference on Robotics and Automation, Stockholm, Sweden, 16–21 May 2016; pp. 5128–5135.
4. Manolakis, D.; Marden, D.; Shaw, G.A. Hyperspectral Image Processing for Automatic Target Detection Applications. *Linc. Lab. J.* **2003**, *14*, 79–116.
5. Chiang, S.S.; Chang, C.I.; Ginsberg, I.W. Unsupervised Target Detection in Hyperspectral. *IEEE Trans. Geosci. Remote Sens.* **2001**, *39*, 1380–1391. [CrossRef]
6. Seydi, S.T.; Hasanlou, M. A new land-cover match-based change detection for hyperspectral imagery. *Eur. J. Remote Sens.* **2017**, *50*, 517–533. [CrossRef]
7. Donoho, D.L.; Johnstone, I.; Stine, B.; Piatetsky-shapiro, G. High-Dimensional Data Analysis: The Curses and Blessings of Dimensionality. *AMS Math. Chall. Lect.* **2000**, *1*, 1–33.
8. Hughes, G. On the Mean Accuracy of Statistical Pattern Recognizers. *IEEE Trans. Inf. Theory* **1968**, *14*, 55–63. [CrossRef]
9. Lee, C.; Landgrebe, D.A. Analyzing High Dimensional Multispectral Data. *IEEE Trans. Geosci. Remote Sens.* **1993**, *31*, 792–800. [CrossRef]
10. Windrim, L.; Ramakrishnan, R.; Melkumyan, A.; Murphy, R.J. A Physics-Based Deep Learning Approach to Shadow Invariant Representations of Hyperspectral Images. *IEEE Trans. Image Process.* **2018**, *27*, 665–677. [CrossRef]
11. Demarchi, L.; Canters, F.; Cariou, C.; Licciardi, G.; Chan, J.C.W. Assessing the performance of two unsupervised dimensionality reduction techniques on hyperspectral APEX data for high resolution urban land-cover mapping. *ISPRS J. Photogramm. Remote Sens.* **2014**, *87*, 166–179. [CrossRef]
12. Bishop, C.M. *Pattern Recognition and Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2006.
13. Murphy, R.; Schneider, S.; Monteiro, S. Mapping Layers of Clay in a Vertical Geological Surface Using Hyperspectral Imagery: Variability in Parameters of SWIR Absorption Features under Different Conditions of Illumination. *Remote Sens.* **2014**, *6*, 9104–9129. [CrossRef]
14. Heege, H.J. *Precision in Crop Farming*; Springer: Berlin/Heidelberg, Germany, 2015; p. 109.
15. Du, Q.; Member, S. Modified Fisher's Linear Discriminant Analysis for Hyperspectral Imagery. *IEEE Geosci. Remote Sens. Lett.* **2007**, *4*, 503–507. [CrossRef]
16. Kuo, B.C.; Li, C.H.; Yang, J.M. Kernel Nonparametric Weighted Feature Extraction for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2009**, *47*, 1139–1155. [CrossRef]
17. Chen, Y.; Lin, Z.; Zhao, X.; Wang, G.; Gu, Y. Deep Learning-Based Classification of Hyperspectral Data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 2094–2107. [CrossRef]
18. Cheriyadat, A.; Bruce, L. Why principal component analysis is not an appropriate feature extraction method for hyperspectral data. *Geosci. Remote Sens.* **2003**, 3420–3422. [CrossRef]
19. Rodarmel, C.; Shan, J. Principal Component Analysis for Hyperspectral Image Classification. *Surv. Land Inf. Sci.* **2002**, *62*, 115–122.
20. Chiang, S.S.; Chang, C.I.; Ginsberg, I.W. Unsupervised hyperspectral image analysis using independent component analysis. *IEEE Trans. Geosci. Remote Sens.* **2000**, *7*, 3136–3138.
21. Cheriyadat, A.M. Unsupervised feature learning for aerial scene classification. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 439–451. [CrossRef]

22. Windrim, L.; Ramakrishnan, R.; Melkumyan, A.; Murphy, R.J. Hyperspectral CNN Classification with Limited Training Samples. In Proceedings of the British Machine Vision Conference (BMVC), London, UK, 4–7 September 2017.
23. Shah, C.A.; Arora, M.K.; Varshney, P.K. Unsupervised classification of hyperspectral data: An ICA mixture model based approach. *Int. J. Remote Sens.* **2004**, *25*, 481–487. [CrossRef]
24. Filho, A.G.D.S.; Frery, A.C.; Araújo, C.C.D.; Alice, H.; Cerqueira, J.; Loureiro, J.A.; Lima, M.E.D.; Oliveira, M.D.G.S.; Horta, M.M. Hyperspectral images clustering on reconfigurable hardware using the k-means algorithm. In Proceedings of the 16th Symposium on IEEE Integrated Circuits and Systems Design, Sao Paulo, Brazil, 8–11 September 2003; pp. 99–104. [CrossRef]
25. Hinton, G.E.; Salakhutdinov, R.R. Reducing the dimensionality of data with neural networks. *Science* **2006**, *313*, 504–507. [CrossRef]
26. Kramer, M.A. Nonlinear principal component analysis using autoassociative neural networks. *AIChE J.* **1991**, *37*, 233–243. [CrossRef]
27. Bourlard, H.; Kamp, Y. Auto-association by multilayer perceptrons and singular value decomposition. *Biol. Cybern.* **1988**, *59*, 291–294. [CrossRef]
28. Licciardi, G.; Frate, F.D.; Duca, R. Feature reduction of hyperspectral data using autoassociative neural networks algorithms. In Proceedings of the 2009 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Cape Town, South Africa, 12–17 July 2009; pp. 176–179.
29. Licciardi, G.; Marpu, P.R.; Chanussot, J.; Member, S.; Benediktsson, J.A. Linear Versus Nonlinear PCA for the Classification of Hyperspectral Data Based on the Extended Morphological Profiles. *Geosci. Remote Sens. Lett. IEEE* **2012**, *9*, 447–451. [CrossRef]
30. Tao, C.; Pan, H.; Li, Y.; Zou, Z. Unsupervised spectral–spatial feature learning with stacked sparse autoencoder for hyperspectral imagery classification. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 2438–2442. [CrossRef]
31. Mou, L.; Ghamisi, P.; Zhu, X.X. Unsupervised spectral–spatial feature learning via deep residual Conv–Deconv network for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 391–406. [CrossRef]
32. Windrim, L.; Melkumyan, A.; Murphy, R.; Chlingaryan, A.; Nieto, J. Unsupervised Feature Learning for Illumination Robustness. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 4453–4457.
33. Chang, C.I. An Information-Theoretic Approach to Spectral Variability, Similarity, and Discrimination for Hyperspectral Image Analysis. *IEEE Trans. Inf. Theory* **2000**, *46*, 1927–1932. [CrossRef]
34. Kruse, F.A.; Lefkoff, A.B.; Boardman, J.W.; Heidebrecht, K.B.; Shapiro, A.T.; Barloon, P.J.; Goetz, A.F.H. The Spectral Image Processing System (SIPS) Interactive Visualization and Analysis of Imaging Spectrometer Data. *Remote Sens. Environ.* **1993**, *44*, 145–163. [CrossRef]
35. Yuhas, R.; Goetz, A.F.H.; Boardman, J.W. Descrimination among semi-arid landscape endmembers using the Spectral Angle Mapper (SAM) algorithm. In Proceedings of the Summaries of the Third Annual JPL Airborne Geoscience Workshop, Pasadena, CA, USA, 1–5 June 1992; Volume 1, pp. 147–149.
36. Hecker, C.; Meijde, M.V.D.; Werff, H.V.D.; Meer, F.D.V.D. Assessing the Influence of Reference Spectra on Synthetic SAM Classification Results. *IEEE Trans. Geosci. Remote Sens.* **2008**, *46*, 4162–4172. [CrossRef]
37. Hyperspectral and Colour Imaging. Available online: https://sites.google.com/site/hyperspectralcolorimaging/dataset (accessed on 2 February 2017).
38. Ramakrishnan, R. Illumination Invariant Outdoor Perception. Ph.D. Thesis, University of Sydney, Sydney, Australia, 2016.
39. Clark, R.; Swayze, G.; Wise, R.; Live, K.; Hoefen, T.; Kokaly, R.; Sutley, S. *USGS Digital Spectral Library splib06a: U.S. Geological Survey Data Series 231*; U.S. Geological Survey: Denver, CO, USA, 2007.
40. Gueymard, C.A. Parameterized Transmittance Model for Direct Beam and Circumsolar Spectral Irradiance. *Sol. Energy* **2001**, *71*, 325–346. [CrossRef]
41. Theodoridis, S.; Koutroumbas, K. *Recognition Pattern*; Academic Press: San Diego, CA, USA, 1998.
42. Lin, T.H.; Li, H.T.; Tsai, K.C. Implementing the Fisher's Discriminant Ratio in a k -Means Clustering Algorithm for Feature Selection and Data Set Trimming. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 76–87. [CrossRef]
43. Wang, S.; Li, D.; Song, X.; Wei, Y.; Li, H. A feature selection method based on improved fisher's discriminant ratio for text sentiment classification. *Expert Syst. Appl.* **2011**, *38*, 8696–8702. [CrossRef]

44. Hubert, L.; Arabie, P. Comparing partitions. *J. Classif.* **1985**, *2*, 193–218. [CrossRef]
45. Rand, W.M. Objective Criteria for the Evaluation of Clustering Methods. *J. Am. Stat. Assoc.* **1971**, *66*, 846–850. [CrossRef]
46. Yeung, K.Y.; Ruzzo, W.L. Principal component analysis for clustering gene expression data. *Bioinformatics* **2001**, *17*, 763–774. [CrossRef]
47. Van Rijsbergen, C.J. *Information Retrieval*; Dept. of Computer Science, University of Glasgow: Glasgow, UK, 1979; Volume 14.
48. Lui, D.C.; Nocedal, J. On the limited memory BFGS method for large scale optimization. *Math. Program.* **1989**, *45*, 503–528.
49. Bengio, Y.; Lamblin, P.; Popovici, D.; Larochelle, H. Greedy Layer-Wise Training of Deep Networks. *Adv. Neural Inf. Process. Syst.* **2007**, *19*, 153.
50. Windrim, L.; Melkumyan, A.; Murphy, R.J.; Chlingaryan, A.; Ramakrishnan, R. Pretraining for Hyperspectral Convolutional Neural Network Classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 2798–2810. [CrossRef]
51. Macqueen, J. Some Methods for Classification and Analysis of Multivariate Observations. In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, CA, USA, 27 December 1965; Volume 1, pp. 281–297.