Segmented Autoencoders for Unsupervised Embedded Hyperspectral Band Selection

1st Julius Tschannerl
Centre for Signal and Image Processing
University of Strathclyde
Glasgow, UK
julius.tschannerl@strath.ac.uk

2nd Jinchang Ren
Centre for Signal and Image Processing
University of Strathclyde
Glasgow, UK
jinchang.ren@strath.ac.uk

3rd Jaime Zabalza
Centre for Signal and Image Processing
University of Strathclyde
Glasgow, UK
jaime.zabalza@strath.ac.uk

4th Stephen Marshall
Centre for Signal and Image Processing
University of Strathclyde
Glasgow, UK
stephen.marshall@strath.ac.uk

***Abstract*—One of the major challenges in hyperspectral imaging (HSI) is the selection of the most informative wavelengths within the vast amount of data in a hypercube. Band selection can reduce the amount of data and computational cost as well as counteracting the negative effects of redundant and erroneous information. In this paper, we propose an unsupervised, embedded band selection algorithm that utilises the deep learning framework. Autoencoders are used to reconstruct measured spectral signatures. By putting a sparsity constraint on the input weights, the bands that contribute most to the reconstruction can be identified and chosen as the selected bands. Additionally, segmenting the input data into several spectral regions and distributing the number of desired bands according to a density measure among these segments, the quality of the selected bands can be increased and the computational time reduced by training several autoencoders. Results on a benchmark remote sensing HSI dataset show that the proposed algorithm improves classification accuracy compared to other state of the art band selection algorithms and thereby builds the basis for a framework of embedded band selection in HSI.**

***Index Terms*—Hyperspectral imaging, autoencoder, band selection.**

I. Introduction

Identifying the most informative wavelengths of a hyperspectral imaging (HSI) dataset and eliminating redundancies whilst simultaneously retaining all relevant information is one of the biggest challenges in HSI data processing. As opposed to closely related feature extraction techniques, that generate new features by e.g. linear combinations or subspace projections, band selection has the significant advantage of retaining information about the process that generated the data and allows for physical interpretation. Contextual subject knowledge about the composition of the imaged objects can help identify relevant spectral regions but only delivers possible solutions for specific applications. The method we propose aims to provide a framework for a generalised approach to hyperspectral band selection without any prior knowledge of the imaged subject and independent of the subsequent data analysis application.

Feature selection can be classified in three categories: *Wrapper*, *filter* and *embedded* methods. Wrapper methods are characterised by evaluating the quality of a selected feature subset by the data analysis algorithm chosen for the specific application, e.g. decision trees [1]. They tend to deliver the best results for the given task but lack generalisability and are often computationally very expensive. Filter methods in contrast define a substitute criterion to evaluate feature subsets and are therefore much less computationally expensive [2]. The third category, embedded methods, differs in the way that it incorporates a feature selection mechanism into the definition of a machine learning algorithm. Similar to wrapper methods, they tend to overfit for the given learning algorithm but are far less computationally expensive [3].

Popular hyperspectral band selection techniques employ similarity measures such as mutual information to determine the most informative bands. The criterion of minimal redundancy and maximum relevance (mRMR) introduced by Peng et. al [4] is used to select bands that best describe class labels by maximising the mutual information between labels and bands. This algorithm therefore requires ground truth data that is not always available. The maximum information and minimum redundancy (MIMR) criterion by Feng et. al [5] can identify the least redundant and most informative subset in an unsupervised manner by maximising the entropy of the individual bands and minimising the mutual information between them. Both algorithms define a substitute criterion and can therefore be classified as filter methods. With recent advances in the field of deep learning, Zhan et. al [6] propose a wrapper method that utilises a convolutional neural network (CNN) to classify the HSI data. Band subsets are generated by segmenting the spectral content into several regions and calculating a newly defined measure called the distance density (DD) for each of the segments. Based on the DD, a different number of bands is selected from each segment and the final subset is evaluated by the CNN. Even though the CNN is optimally designed so it does not need to be retrained for every subset, the algorithm still suffers from high computational cost due to repeated evaluations. Embedded band selection algorithms incorporate the subset selection into the training of the learning algorithm. Yang et. al [7] have adopted the popular embedded feature selection least absolute shrinkage and selection operator (LASSO) for hyperspectral data with good results mainly for a higher number of selected bands. LASSO, however, can only exploit linear relationships between the input features. The recent research focus on deep learning algorithms, and autoencoders (AEs) in particular, in various fields of machine learning led us to investigate the usage of deep learning algorithms for embedded band selection as they are able to handle any sort of input data and have a strong capability of dealing with non-linear relationships. An AE is in the simplest form a neural network with an input and output layer as well as one hidden layer. The aim is to reconstruct the input at the output, hence the hidden layer can be interpreted as an encoded version of the input. Chandra et. al [8] introduced a feature selection algorithm based on AEs. By masking input features, i.e. setting their input weight to 0, and subsequently comparing the reconstruction error between each feature being present or not present, the features that generate the largest difference in the error are considered to be most relevant. Han et. al [9] have explored the possibility of AEs for feature selection for facial recognition in digital image data. By putting a sparsity constraint on the input weights, it is possible to identify the features that contribute most to the reconstruction. Zabalza et. al [10] have utilised a segmented stacked AE (S-SAE) for hyperspectral feature extraction by the hidden layer as a lower dimensional representation. By segmenting the spectral content into several regions and training multiple SAEs, the performance could be optimised and the computational cost for extracting features from an already trained network decreased. In this paper, we are combining the idea of segmentation for feature extraction with the concept of distance density and the idea to utilise input weights of AEs to select most significant input features to generate a framework for unsupervised, embedded hyperspectral band selection.

II. Proposed algorithm

The proposed algorithm consists of several steps. At first, the hyperspectral data is analysed and segmented into several spectral regions. by calculating the distance density for each segment, the number of desired bands can be distributed accordingly among these segments. For each segment, an autoencoder with a sparsity constraint on the input weights is trained and the corresponding number of input bands with the highest weights are selected.

A. AE based band selection

A basic AE model is a special feedforward neural network with one input layer and two fully connected layers. Its purpose is to reconstruct the input at the output layer by learning a lower dimensional, abstract representation of the data at the hidden layer. We define a simple autoencoder based on [9]. For an input matrix $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_m\}^T \in \mathbb{R}^{m \times d}$, where m is the number of input samples and d is the dimensionality of the input, an AE is defined by two functions. The encoder function $\mathbf{f}_i = \sigma_1(\mathbf{W}^1\mathbf{x}_i + \mathbf{b}^1)$ and the decoder function that reproduces the input matrix $\hat{\mathbf{x}}_i = \sigma_2(\mathbf{W}^2\mathbf{f}_i + \mathbf{b}^2)$. σ_1 and σ_2 are the activation functions of the hidden and output layer respectively, $\mathbf{W}^i$ represents the weight matrices and $\mathbf{b}_i$ the bias vectors for each layer. $\mathbf{w}^l_{ij}$ denotes the weight of the connection between the i-th node in the l-th layer and the j-th node in the $(l+1)$-th layer and $\mathbf{b}^l_i$ denotes the additive bias term of the i-th node in the l-th layer.

For training, we can define the AE as a loss function $\mathcal{J}(\Theta)$ of the difference between the input and output with parameter $\Theta = \{\mathbf{W}^1, \mathbf{W}^2, \mathbf{b}^1, \mathbf{b}^2\}$.

$$\mathcal{J}(\Theta) = \frac{1}{2m}\|\mathbf{X} - \hat{\mathbf{X}}\|_F \tag{1}$$

where $\|\mathbf{A}\|_F$ is the Frobenius norm of matrix $\mathbf{A}$. To realise feature selection, [9] suggest to add a row-sparse regularisation term on the input weight matrix $\mathbf{W}^1$ which is realised by the $L_{2,1}$ norm:

$$\|\mathbf{W}^1\|_{2,1} = \sum_{i=1}^{d}\sqrt{\sum_{j=1}^{h}(\mathbf{W}^1_{ij})^2} \tag{2}$$

The i-th row $\mathbf{w}_i^T$ of $\mathbf{W}^1$ corresponds to the i-th feature and $\|\mathbf{w}_i\|$ gives indication on the contribution of the i-th feature to the reconstruction. The resulting loss function is defined as

$$\mathcal{J}(\Theta) = \frac{1}{2m}\|\mathbf{X} - \hat{\mathbf{X}}\|_F + \alpha\|\mathbf{W}^1\|_{2,1} + \frac{\beta}{2}\sum_{i=1}^{2}\|\mathbf{W}^i\|_F \tag{3}$$

where α is a trade-off parameter between the reconstruction loss and the sparsity regularisation. An additional weight decay term is added with β being the penalty parameter. This term prevents overfitting and enforces convergence of the optimisation. After optimisation, the bands are indicated by the norms of the columns of the input weight matrix $\mathbf{W}^1 = (\mathbf{w}_1\mathbf{w}_2...\mathbf{w}_d)$ for d input bands, where $max\ |\mathbf{w}_i|$ indicates band i has the highest relevance.

In a hyperspectral dataset, each pixel's spectrum can be used as an input to the AE. The selected features from the defined AE represent those bands, that are most relevant to the reconstruction of the spectrum and can be interpreted as the most informative bands. The functionality is depicted in Figure 1.

B. Segmentation of spectral regions

The spectral region covered by the utilised sensor can usually be divided into several logical segments and each of these regions contain a different amount of information about the dataset. Other algorithms, such as segmented principal component analysis [11] have adopted this concept successfully in the past. These segments are commonly generated by looking at the correlation matrix of the spectral bands. More information about the segmentation will be given in Section III. Once the dataset is segmented into spectral regions, one AE for each segment can be trained and the resulting bands

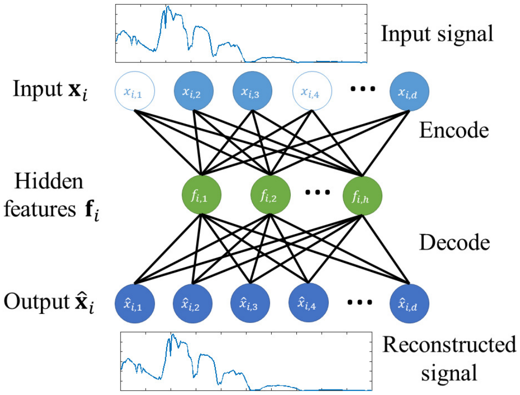

Input signal
Input $\mathbf{x}_i$
$x_{i,1}$ $x_{i,2}$ $x_{i,3}$ $x_{i,4}$ ··· $x_{i,d}$
Encode
Hidden features $\mathbf{f}_i$
$f_{i,1}$ $f_{i,2}$ ··· $f_{i,h}$
Decode
Output $\hat{\mathbf{x}}_i$
$\hat{x}_{i,1}$ $\hat{x}_{i,2}$ $\hat{x}_{i,3}$ $\hat{x}_{i,4}$ ··· $\hat{x}_{i,d}$
Reconstructed signal

Fig. 1: Schematic of AE band selection. Input bands with the highest weights contribute most to the reconstruction of the signal

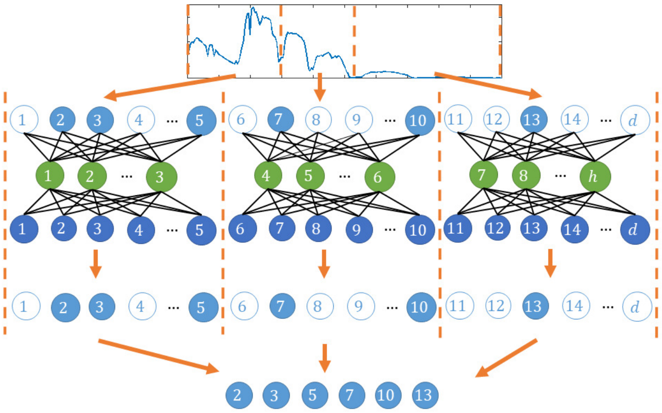

1 2 3 4 ··· 5 | 6 7 8 9 ··· 10 | 11 12 13 14 ··· d
1 2 ··· 3 | 4 5 ··· 6 | 7 8 ··· h
1 2 3 4 ··· 5 | 6 7 8 9 ··· 10 | 11 12 13 14 ··· d
1 2 3 4 ··· 5 | 6 7 8 9 ··· 10 | 11 12 13 14 ··· d
2 3 5 7 10 13

Fig. 2: Schematic of S-AE band selection. The input data is segmented in to spectral regions and the results of the independent AEs are concatenated.

of each segment are simply concatenated. This process is visualised in Figure 2

As mentioned above, each spectral segment likely has a different amount of information necessary for the reconstruction. To account for this, the concept of distance density from [6] is adopted here. The distance density dd_i for segment i with m samples and n bands is defined as:

$$dd_i = \frac{1}{n-1}\sum_{j=1}^{n-1} d_j;\ d_j = \sum_{k=1}^{m} |r_{j+1k} - r_{jk}| \tag{4}$$

where d_j is the absolute difference between the reflectance values r_{jk} of two adjacent bands j and $j+1$ in sample k.

The number of bands n_{b_i} for the i-th segment can then be calculated by:

$$n_{b_i} = \frac{dd_i}{\sum_{i=1}^{s} dd_i} \times n_b \tag{5}$$

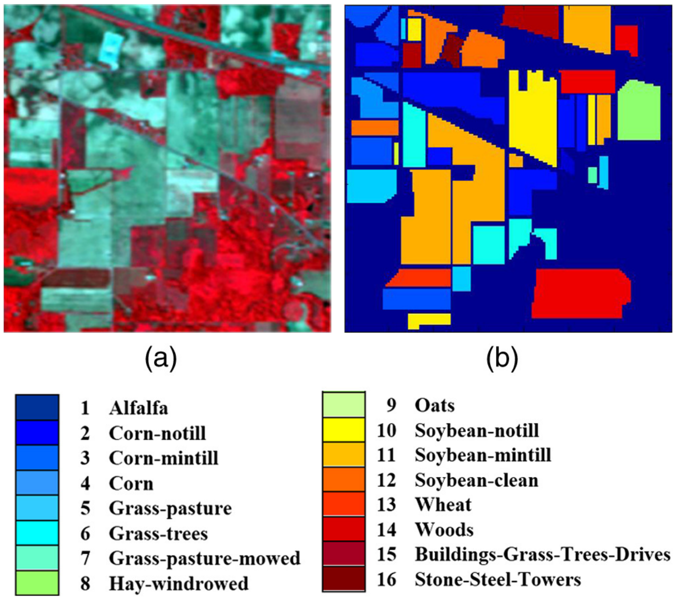


Fig. 3: Indian pines dataset with class description. (a) false colour representation (b) ground truth.

where s is the number of segments and n_b the total number of desired bands.

According to this calculation, spectral segments with a high information density yield more selected bands to the final subset than segments with a low density.

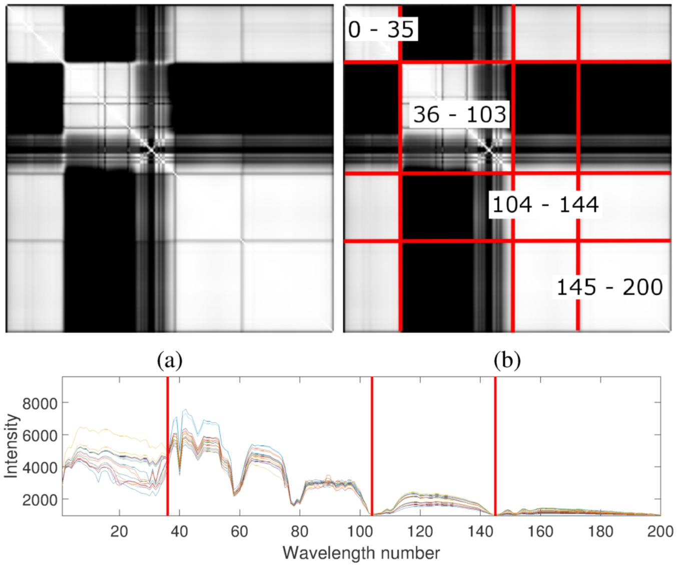
III. Experimental Results

The algorithm was tested on the publicly available remote sensing Indian Pines HSI dataset. It was collected by the AVIRIS sensor at the Indian Pines testsite in northwest Indiana. It comprises mainly of agriculture and some natural vegetation divided into 16 classes. Removing noisy water absorption bands, it consists of 200 spectral bands covering a range from 400 - 2500nm at 145 x 145 pixels. It is depicted in Figure 3.

A. Segmentation

Choosing the right segments has significant impact on the classification performance [10]. Other than in [6] where the overall spectral region is divided into several equally sized segments, we are trying to identify logical regions dependent on the specific dataset. The correlation matrix can help with that. It is depicted in Figure 4. Alongside, the mean spectra of all classes are shown to further verify the choice of regions. Only a few segmentation options are possible and the segments here are chosen by manual inspection as they produce the best results.

Based on the distance density from Equation 5, the number of bands for each segment depending on the total number of desired bands can be calculated. Examples for the distribution between the segments for different number of bands can be seen in Table I. One can see that in the Indian Pines dataset, the first two segments contain significantly more information than



(a)

(b)

(c)

Fig. 4: (a) Correlation matrix with (b) selected segments and (c) the mean spectra per class with according segments for the Indian Pines dataset.

TABLE I: Number of bands selected for each segment

# Bands	0 - 35	36 - 103	104 - 144	145 - 200
10	3	5	1	0
30	10	16	3	1
100	34	54	9	3

the last two. Judging from the spectral signatures in Figure 4c, the first two segments display much more variance between the classes and between adjacent bands than the other segments.

B. Performance analysis

To assess the performance of the proposed algorithm, it has been tested in various configurations. As a means to evaluate the influence of the segmentation, AE Band Selection (AEBS) without segmentation was performed and compared with the results of the Segmented AE Band Selection (S-AEBS). Based on tuning, the parameters α and β of S-AEBS were set to 0.0001 and 0.1 respectively. The hidden layer consists of 1/2 the number of nodes as the input layer, every S-AE is terminated after 3000 iterations and the AE without segmentation takes longer to converge and is therefore terminated after 6000 iterations. σ_1 is set to a sigmoid function and σ_2 is set to the identity, according to [9]. Additionally, based on the segmentation and distance density, random features of each segment were selected, labelled S-RandBS. To benchmark the overall performance, the algorithm was compared with state of the art unsupervised feature selection algorithms including Ward's Linkage strategy using Mutual Information (WaLuMI) [12] and MIMR optimised by Clonal Selection Algorithm (MIMR-CSA) [5]. WaLuMI clusters the spectral bands based on mutual information and selects representatives of each cluster, whereas MIMR-CSA tries to find band subsets that have individually maximum entropy and minimal mutual information between the bands. The subset candidates are generated using CSA, which iteratively clones and mutates subsets based on their fitness over several generations. MIMR-CSA was applied using 100 iterations, a population size $s = 50$ and the mutation probability factor n_m, the selection probability factor n_s, the number of clones n_{cl} and number of displaced antibodies d were set to 5, 0.5, 2500 and 5 respectively as specified in [5].

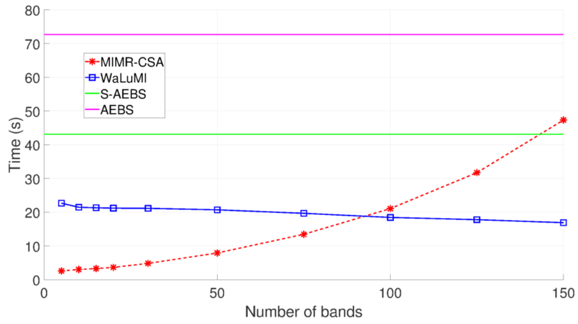


Fig. 5: Time consumption for all algorithms on the Indian Pines dataset.

1) Computational complexity: All algorithms were run on an Intel core i5 @ 3.2 GHz with 16GB RAM. S-AEBS and AEBS are both implemented in the tensorflow framework for Python, MIMR-CSA is implemented in Matlab and for WaLuMI, the C++ implementation provided in [12] is used. Because of the different implementations and runtime environments, the times cannot be directly compared. However, general trends and structural differences of the algorithms can be identified. Runtime measurements are shown in Figure 5. As stated in [5], MIMR-CSA has a quadratic runtime with respect to the number of desired features. The runtime however is significantly reduced when mutual information and entropy are pre-calculated into lookup tables. WaLuMI, as stated in [12], accounts most of its computational time to the pre-clustering step, where the mutual information is calculated. Since the clustering step does not significantly impact the computational time, the number of desired features does not change the time consumption noticeably. AEBS and S-AEBS both are trained on all available bands, even if S-AEBS segments the spectral region. The band selection takes place in a subsequent process with linear runtime where all bands are evaluated. Therefore both AEBS and S-AEBS show the same runtime for every number of features selected. As mentioned in Section III-B, AEBS takes more time to converge and is therefore given twice as many iterations. Even though MIMR-CSA outperforms the other algorithms for lower number of features, it still requires the pre-calculation of mutual information and entropy, which can last up to several days. S-AEBS performs faster than AEBS, even though 4 different AEs are trained. This is partly due to the fact that it requires less iterations and also that less input bands results in a less complex AE. Given that these

TABLE II: Class-wise accuracies for individual algorithms on the Indian Pines dataset selecting 30 bands

Class	WaLuMI	MIMR-CSA	S-RandBS	AEBS	S-AEBS
1	60.5±15.8	58.0±12.7	54.0±14.9	**68.9±11.2**	64.8±14.0
2	**77.2±2.1**	72.6±3.1	73.4±4.2	61.3±5.0	74.8±2.5
3	63.3±1.7	64.3±3.3	60.2±6.5	40.8±3.9	**64.4±4.1**
4	65.2±7.1	59.5±8.2	58.3±7.2	52.8±7.3	**67.8±7.7**
5	87.7±4.3	**90.9±2.1**	89.5±3.0	86.7±4.5	86.9±3.7
6	93.3±1.2	**94.7±2.0**	93.6±2.0	94.4±1.3	92.3±2.1
7	**75.2±13.1**	73.4±14.1	64.4±20.3	55.8±22.4	72.9±18.8
8	95.6±2.5	97.0±1.6	96.8±1.6	**97.3±1.3**	96.8±1.7
9	**45.6±9.9**	42.9±15.2	37.1±13.8	26.4±18.2	30.9±16.3
10	**75.4±1.5**	70.6±4.9	70.8±5.4	54.3±9.0	74.3±4.5
11	80.5±1.8	81.4±1.6	80.1±2.2	77.9±2.9	**82.1±2.1**
12	70.7±2.2	70.6±3.0	67.8±5.8	38.7±6.5	**75.3±3.5**
13	93.6±3.9	95.7±2.8	94.6±3.5	**96.2±3.5**	95.5±2.7
14	95.0±1.9	95.3±0.9	94.7±1.7	95.5±2.1	**95.6±1.3**
15	48.0±2.2	50.0±6.0	**50.8±6.7**	44.0±4.1	49.5±5.6
16	84.1±4.5	82.7±6.4	**86.0±5.6**	81.3±6.6	84.4±5.8
OA	79.9±0.3	79.4±0.8	78.5±2.1	71.1±1.7	**80.4±0.8**
AA	**75.7±1.1**	75.0±1.8	73.3±3.1	67.0±3.0	75.5±1.9
Kappa	77.0±0.3	76.4±1.0	75.3±2.5	66.7±2.0	**77.5±0.9**

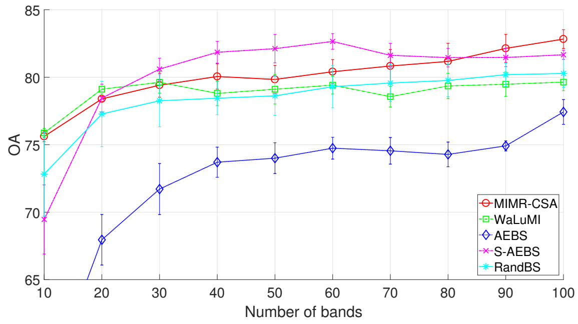


Fig. 6: Classification accuracies for different algorithms on the Indian Pines dataset.

AEs are independent from each other, there is potential for a straight forward parallel CPU implementation which can significantly further reduce this time.

2) Classification performance: To evaluate the quality of the selected bands, the reduced datasets were classified using a support vector machine (SVM) with a radial basis function (RBF) kernel whose parameters C and γ were tuned with a grid search and five-fold cross validation. 10% of pixels of each class was randomly selected for training, and the rest to test the classifier. Since the AE optimisation is done with random initialisation, each training process will produce a slightly different band subset. To account for this, 30 AEs were trained in both AEBS and S-AEBS and each of these subsequently classified with 5 SVMs with different training and testing samples, resulting in 150 runs. Class-wise accuracies can be seen in Table II. Due to very different number of samples available for each class, accuracies may vary strongly. Overall, S-AEBS outperforms all algorithms in terms of Overall Accuracy (OA) and Cohen's Kappa coefficient and is only slightly outperformed by WaLuMI in terms of class-wise average accuracy (AA) by 0.2%.

In Fig. 6, the OAs of the different algorithms with respect to the number of bands selected were compared. One can see that while AEBS performs consistently the worst, the introduction of the segmentation significantly increases the accuracy. This is also affirmed by the fact, that the random selection of bands within the segmentation outperforms the standard AEBS. The random selection however quickly reaches its limits and has, as shown in Table II, a relatively high variance due to its randomness. While WaLuMI and MIMR-CSA outperform S-AEBS for a low number of selected bands, S-AEBS quickly overtakes both. We believed that the input weight of the bands cannot be directly mapped to their importance, which is why selecting only few bands is not working very well for this approach. For a higher number of bands (80+), MIMR-CSA seems to perform better. Overall, S-AEBS seems to provide a good foundation for embedded hyperspectral feature selection, that may outperform state-of the art algorithms with further modifications.

IV. Conclusion and Future Work

In this paper, we proposed an autoencoder design for embedded hyperspectral band selection. By putting a sparsity constraint on the input weights, spectral bands that contribute most to the reconstruction can be identified. Combining this approach with a segmentation of the spectral region and training several AEs results in a faster and better band selection than a regular AE. This forms the basis for AE band selection that can compete with state of the art algorithms. While the time consumption of training several AEs is relatively high, CPU and GPU parallelisation of the S-AE training can be utilised to speed up the selection and improve the performance. Furthermore, comparable algorithms rely on the pre-calculation of information theoretic measures that can consume a considerable amount of time in itself. Future work may also include an automatic segmentation procedure. Further research into the optimisation of AE configuration and training can potentially improve the selection performance and provide a band selection approach that outperforms state-of-the-art algorithms in terms of computational complexity as well as band selection quality.

References

[1] Y. Zhang, S. Wang, P. Phillips, and G. Ji, "Binary PSO with mutation operator for feature selection using decision tree applied to spam detection," *Knowledge-Based Syst.*, vol. 64, pp. 22–31, 2014.

[2] H. G. Vijouyeh and G. Taskin, "A comprehensive evaluation of feature selection algorithms in hyperspectral image classification," in *2016 IEEE Int. Geoscience and Remote Sensing Symp. (IGARSS)*, 2016, pp. 489–492.

[3] J. C. Hernandez Hernandez, B. Duval, and J.-K. Hao, “A genetic embedded approach for gene selection and classification of microarray data,” in *Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics: 5th Europ. Conf.*, 2007, pp. 90–101.

[4] H. Peng, F. Long, and C. Ding, “Feature selection based on mutual information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, 2005.

[5] J. Feng, L. Jiao, F. Liu, T. Sun, and X. Zhang, “Unsupervised feature selection based on maximum information and minimum redundancy for hyperspectral images,” *Pattern Recognit.*, vol. 51, pp. 295–309, 2016.

[6] Y. Zhan, D. Hu, H. Xing, and X. Yu, “Hyperspectral Band Selection Based on Deep Convolutional Neural Network and Distance Density,” *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 12, pp. 2365–2369, 2017.

[7] D. Yang and W. Bao, “Group Lasso-Based Band Selection for Hyperspectral Image Classification,” *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 12, pp. 2438–2442, 2017.

[8] B. Chandra and R. K. Sharma, “Exploring autoencoders for unsupervised feature selection,” in *2015 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2015, pp. 1–6.

[9] K. Han, C. Li, and X. Shi, “Autoencoder inspired unsupervised feature selection,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, forthcoming.

[10] J. Zabalza, J. Ren, J. Zheng, H. Zhao, C. Qing, Z. Yang, P. Du, and S. Marshall, “Novel segmented stacked autoencoder for effective dimensionality reduction and feature extraction in hyperspectral imaging,” *Neurocomputing*, vol. 185, pp. 1–10, 2016.

[11] X. Jia and J. A. Richards, “Segmented principal components transformation for efficient hyperspectral remote-sensing image display and classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 37, no. 1 II, pp. 538–542, 1999.

[12] A. Martínez-Usó, F. Pla, J. M. Sotoca, and P. García-Sevilla, “Clustering-based hyperspectral band selection using information measures,” *IEEE Trans. Geosci. Remote Sens*, vol. 45, no. 12, pp. 4158–4171, 2007.