A Study of Deep Learning Approaches and Loss Functions for Abundance Fractions Estimation

Vaibhav Lodhi
IIT Kharagpur
Kharagpur, India
vaibhav.lodhi@gmail.com

Arindam Biswas
IIT Kharagpur
Kharagpur, India

Debashish Chakravarty
IIT Kharagpur
Kharagpur, India

Pabitra Mitra
IIT Kharagpur
Kharagpur, India

Abstract—**Spectral unmixing is one of the post processing operation of hyperspectral image processing. In general, it is observed that the traditional algorithms are not efficient enough to estimate the abundance fractions of endmembers. In recent years, neural network based approaches are competent enough to perform the complex operations for remote sensing applications. After the origination of deep learning, number of deep learning approaches have been proposed and trained using different loss functions for performing several complex operations. Hence, it is necessary to select suitable deep learning model and loss function for the application(s). The objective of this work is to study the suitability of the model which affects the performance of the unmixing operation. In this work, estimation accuracy of different deep learning models and loss functions for spectral unmixing operation using hyperspectral data have been studied. In this work, five deep learning models are implemented and trained using four loss functions. Evaluation of proposed study has been carried using available real hyperspectral datasets. Hence, It is observed from the study that the different deep learning models and loss functions affect the estimation accuracy of spectral abundance fractions. In this work, parallel convolution 1-D model has been performed best among the implemented approaches for estimation of abundance fractions.**

Index Terms—**Abundance Fractions, Deep learning, Hyperspectral Imaging, Loss function, Spectral Unmixing**

I. Introduction

Hyperspectral imaging is one of the popular Earth Observation [1], [2] and Geoimaging techniques. It has been applied in several applications which includes environmental surveys [3], [4], agriculture, [5], [6], geology [7] and so on. This technique includes spatial and spectral information of the scene under observation. In general, it is observed that mixed pixels occur due to the coarse resolution of hyperspectral imagery. Spectral unmixing is one of the vital tasks of hyperspectral image post-processing operation in which measured spectrum of the mixed pixel is decomposed into a collection of endmembers and their corresponding abundance fractions. It consists of two operations, i.e., endmember extraction and abundance fractions estimation of endmembers.

Some of the existing unmixing algorithms for abundance fractions estimation operation are Fully Constrained Least Squares (FCLS) [8], Unconstrained Least Square (UCLS), and Non-Negative constrained Least Square (NNLS). While Vertex component analysis [9] and N-FINDR [10] techniques are used for endmember extraction in unmixing. In general, unmixing depends upon preconsidered mixing model which may be either linear or nonlinear in hyperspectral image processing. Linear Mixing Model (LMM) does not consider multiple scattering and intimate mixture effects [11]. Some algorithms and approaches have been proposed to solve linear unmixing problem [11], [12]. However, LMM is not an actual expression of many real-world scenarios. Many models have been proposed to solve nonlinear mixing which includes Fan model [13], Modified GBM (MGBM) [14], Polynomial Post-Nonlinear model (PPNMM) [15] and Multilinear mixing model (MLM) [16]. However, most of the nonlinear model for abundance fractions estimation are model-dependent, i.e., applicable for certain case. Hence, unmixing results may not be guaranteed, if the considered mixing model does not match with the real scene. In a real scenario, it is not possible to consider prior mixing model of different hyperspectral images.

In recent years, usage of neural network algorithms have been increased in remote sensing applications [17]. Many researchers suggested considerable number of advantages of neural network approaches over other methods. Applications of neural network approaches have been increased due to their ability to learn complex patterns, i.e., no prior assumption is needed about the distributions of dataset [18]. In [19], it is observed that the deep learning models are more accurate for spectral unmixing operation than their traditional counterparts. Supervised deep learning models [20] like stacked encoders and PCAnet have also been used for abundance estimation task. In [21], authors have also been used autoencoders to do blind unmixing operation.

In recent times, there are number of deep learning models and their loss functions have been proposed. The objective of this work is to study the different deep learning models and loss functions for abundance fractions estimation in spectral unmixing operation. In this proposed work, supervised deep learning models have been implemented with more complexity than the shallow neural networks. Consequently, Complexity of the model encodes the information better than the shallow networks for hyperspectral data. Implemented supervised deep learning models for abundance fractions estimation consist of convolution based models as well as Long Short Term Memory (LSTM) units [22] based models. Convolutional neural networks have staple in the field of computer vision [23], producing promising results. LSTM units have also been very effective for sequential data like text. Thus, it produces promising results for text analysis. However, proposed models are supervised learning models, i.e., data-driven models; therefore, data is required for training before abundance fractions estimation operation. In the proposed models, there is no need to assume mixing concept for unmixing operation. Implemented models have been trained for end-to-end mapping, i.e., from spectral information of a pixel to its abundance fractions without considering its mixing model. Experiments on real hyperspectral data sets have been performed using different models and loss functions.

Other sections of the paper have been discussed below. Section II discusses the methodological part of proposed work. Implemented models and loss functions have been discussed in section III. Experiment and results, and conclusion and future scopes have been discussed in section IV and V.

II. Methodology

In this section, methodology for the proposed work has been discussed. For abundance fractions estimation, five deep learning architectures have been proposed and trained using four loss functions which are discussed below. Each proposed model has been trained using labelled data.

Generalized schematic representation of deep learning architecture for the proposed work has been shown in Figure 1. It is observed from the Figure that the single pixel of the hyperspectral dataset has been given as an input to the LSTM and convolutional based feature encoders. After that, it is then followed by fully connected layers to output. Labelled data has been used to train a model. Trained models have been used for estimating the abundance fractions. To obtain the better results, tuning of hyperparameters has been done for each network and dataset. Adam optimiser has been used for training since it is more robust for deeper models with a learning and decay rate of 0.001 and $\beta 1$ = 0.001. Along with, mini-batch training technique has been used with a batch size of 64. In the proposed models, two constraints for abundance fractions, i.e., non-negative and sum up-to-one have been considered. To ensure these constraints, softmax layer has been applied over the output of the last fully connected layer. An expression for the sum-to-one and non-negative constraints has been given below:

$$\sum_{l=1}^{R} \alpha_{il} = 1, \qquad \alpha_{il} \geq 0 \qquad \forall l = 1, ..., R. \tag{1}$$

In above equation, α_{il} represents the abundance fraction of l^{th} endmember of i_{th} pixel and R indicates the number of endmembers.

III. Proposed Models

Five deep learning based models have been proposed for abundance fractions estimation of the endmembers. They are Long Short Term Memory (LSTM) Network, Time distributed LSTM Network, Convolution 1-D Network, Parallel Convolution 1-D Network, and Bidirectional LSTM Network. The proposed models include convolutions and LSTM based feature encoders which are followed by a fully connected layer. The input of the models is a single pixel X_i of size $\lambda * 1$ extracted from the hyperspectral image of dimension $X * Y * \lambda$. Each of the model has been discussed below:

- **Long Short Term Memory (LSTM) Network** For LSTM, value of a single pixel at different wavelengths is taken as a sequential data. So, recurrent neural networks like LSTM network can apply to data. In this model, a series of LSTM cells has been used, followed by the fully connected layers to predict the abundance fractions of each pixel. In this model, we have used the unit size of 198.
- **Time distributed LSTM Network** This model is different from the LSTM network which generates single output. In time distributed LSTM network, each LSTM cell generates output. The output from each of the LSTM cells is sent to the time distributed layer which encodes the output from each cell. Then, it is followed by fully connected layers.
- **Convolution 1-D Network** Each pixel of the hyperspectral data is considered as a spatial data for convolution 1-D network. Convolutional 1-D layers are followed by max-pooling in series to capture the feature of this spatial data. Finally, it is followed by the fully connected layers. For the implementation of convolution layer, we have used the filter size of 25. While size of max pooling and kernal is of 3.
- **Parallel Convolution 1-D Network** In this model, a similar model is made like the convolutional 1-D network, but in this case, there would be three different convolutional 1-D networks are in parallel. Each one has a different filter/kernel size for convolution. Finally, a merged output of parallel networks is sent to a fully connected network. In this model, we have used the kernel size of 3,4 and 5.
- **Bidirectional LSTM Network** In this model, we make use of bi-directional network using LSTM units. Since, data is considered to be sequential. Hence, network tries to learn from forward and backward direction in order to encode more information. Afterwards, output of this bi-directional network sends to a fully connected network.

A. Loss Functions

In general, models are trained using an optimization process that needs a loss function to compute the model error. Loss function plays a vital role in order to reduce the all aspects of model into a single value. Accordingly, candidate solutions have to be ranked and compared. Selection of poor error function produces unsatisfactory results. Hence, four different loss functions are used here in order to train the implement five deep learning based models. Each of the loss function has been discussed below:

- **Mean Square Error Loss** It is the average of the square of the deviations between the estimated $\hat{Y}_i$ and real value Y_i. An expression for the mean square error loss has been given below:

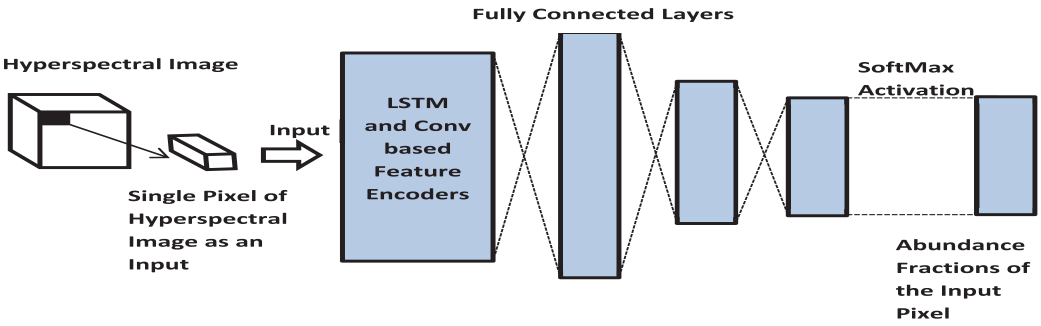
Fully Connected Layers
Hyperspectral Image
Input
LSTM and Conv based Feature Encoders
SoftMax Activation
Single Pixel of Hyperspectral Image as an Input
Abundance Fractions of the Input Pixel

Fig. 1. Generalized Schematic Representation of Deep Learning Architecture for the Proposed Work

$$MSE = \frac{1}{n}\Sigma_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \qquad (2)$$

- **Log Cosh Loss** In Log cosh loss, logarithm of the hyperbolic cosine of the prediction error $(x = Y_i - Yi^{(}i))$ is taken i.e., log(cosh(x)). In general, it is approximately equal to $(x**2)/2$. Logcosh works closely like the mean squared error and will not be strongly affected by the occasional wildly incorrect prediction.
- **Cosine Similarity Loss** Cosine similarity is a measure of similarity between two non-zero vectors. It is an inner product space that measures cosine of the angle between them. In this case, we consider the discrete probability distribution as vectors on which cosine similarity is calculated. Expression for cosine similarity loss has been given below:

$$cos\theta = \frac{A.B}{\parallel A \parallel \parallel B \parallel} = \frac{\Sigma_{i=1}^{n} A_i B_i}{\sqrt{\Sigma_{i=1}^{n} A_i^2}\sqrt{\Sigma_{i=1}^{n} B_i^2}} \qquad (3)$$

  Where, A and B are non-zero vectors.
- **Cross Entropy Loss** The cross-entropy between two probability distributions a and b over the same underlying set of events measures the average number of bits needed to identify an event drawn from the set. If a coding scheme is used i.e., optimised for an unnatural probability distribution b, rather than the true distribution a. For discrete a and b, cross-entropy loss expression has been given below:

$$H(a,b) = -\Sigma_x a(x) log b(x) \qquad (4)$$

B. Evaluation Metrics

Two evaluation criteria have been used here, and they are Root Mean Square Error (RMSE), and cosine similarity. Expressions for the evaluation metrics are given below:

- **RMSE :** This is used to measure the average difference between two values i.e., actual and predicted value.

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}(a-b)^2} \qquad (5)$$

- **Cosine Similarity :** This metric is used to measure the similarity between two non-zero vectors (a and b). It is an inner product space that measures the cosine of the angle between them.

$$cos\theta = \frac{a.b}{\parallel a \parallel \parallel b \parallel} \qquad (6)$$

IV. Experiment and Results

In this section, experiment and results have been demonstrated for the proposed work. All the implemented deep learning models along with four loss functions have been used for experimentation purpose. All the proposed models along with different loss functions have been trained to determine abundance fractions of endmembers. They are tested using popular benchmarked hyperspectral datasets. JasperRidge and Samson hyperspectral datasets have been used as shown in Figure 2 and Figure 4. Both the datasets have different number of endmembers.

JasperRidge [24]–[26] is one of the popular hyperspectral dataset with a spectral resolution of 9.46 nm. Each pixel of the dataset ranges from 380 nm to 2500 nm. Dimension of the dataset includes 100 x 100 pixels with 198 channels. There are four endmembers in this dataset, i.e., tree, soil, water, and road. Spectras of the endmembers have been shown in Figure 3. Similarly, Samson [24]–[26] dataset has also been used to determine the best performing model and loss function. Dimension of the dataset includes 95 x 95 pixels with 156 channels, and it covers the wavelength range from 401 nm to 889 nm. There are three endmembers in this dataset, i.e., soil, tree, and water. Spectras of the endmembers have been shown in Figure 5. This work has been implemented using python tensorflow.

For training purpose, randomly shuffled the pixels and pick out 55% of the pixels along with their labels. After that, trained models are tested with the remaining 45% pixels of the dataset.

Different loss functions have been used for training the models and they are mean square error loss, logcosh loss, cosine loss, and cross-entropy loss function, respectively. Root mean square error (RMSE) and cosine similarity metrics are used to determine the accuracy of the models. RMSE metric represents the difference between the real and predicted value using the spatial distance while the cosine similarity metric calculates the spatial angle between the two values. Table I depicts the RMSE and cosine similarity for all the models using different loss functions on both the datasets. It is observed that most of the models with different loss function have been performing well as indicated by Table I. However, Parallel convolution 1-D model, convolution 1-D has been performing well among the proposed models for estimation of abundance fractions.

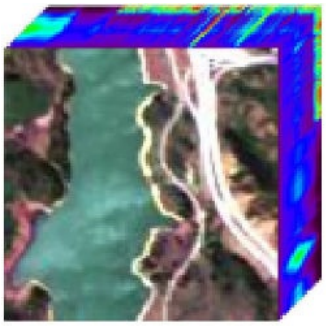

Fig. 2. JasperRidge Dataset

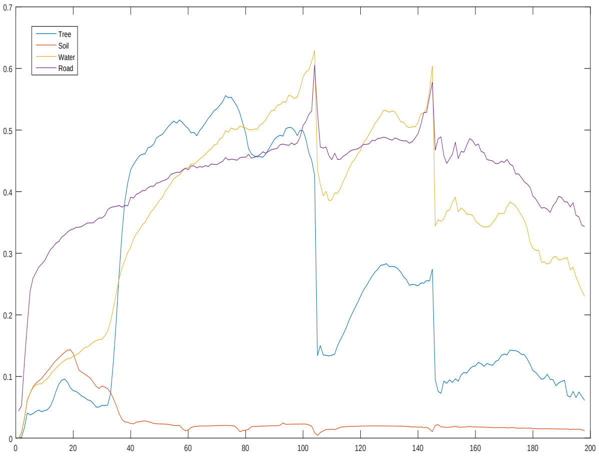

Fig. 3. Endmembers for JasperRidge Dataset

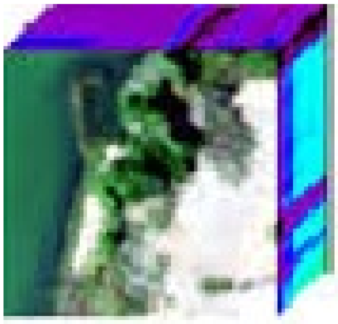

Fig. 4. Samson Dataset

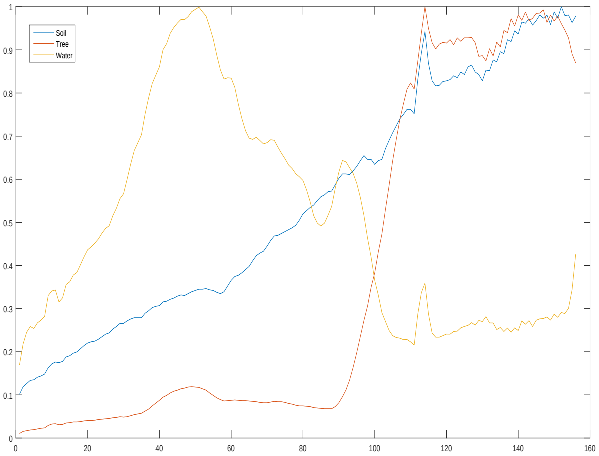

Fig. 5. Endmembers for Samson Dataset

In case of JasperRidge (4 endmembers) hyperspectral dataset, parallel convolution 1-D model with mean square error loss function has been performing well. In this case, values of RMSE and cosine similarity are 0.039 and 0.995. While for Samson (3 endmembers) dataset, parallel convolution 1-D model with cross-entropy loss function has been performing well and the values of RMSE and cosine similarity are 0.028 and 0.998. It is noticed that the parallel convolution 1-D model has been performing best for both the datasets by using different loss functions. This is due to the fact that parallel convolution 1-D model includes several 1D-CNN in parallel to learn data presentation at different scales. As a result, it can effectively learn features from a richer data representation automatically by many layers of convolution and pooling operations. Alongwith, it is also observed that RMSE and cosine error vary by using different loss function. Hence, it can be say that the models and loss functions affect the accuracy for the estimation of abundance fractions of endmembers. Therefore, it is needed to select suitable deep learning model and loss function for estimation of abundance fractions of endmembers.

V. Conclusion

Estimation of spectral abundance fractions is one of the major post-processing tasks of hyperspectral image processing. Herein, five deep learning models are implemented to estimate the abundance fractions. They are trained using four loss functions. After that, comparison analysis has been done among the approaches using different loss functions. For this purpose, JasperRidge and Samson hyperspectral datasets have been used for estimation of abundance fractions. It is observed that the deep learning models with different loss functions have been generated a promising result. However, parallel convolution 1-D model has been performing best

TABLE I
RMSE AND COSINE ERROR FOR THE PROPOSED METHODS

	JasperRidge		Samson	
Model	**RMSE**	**Cosine**	**RMSE**	**Cosine**
	Mean Square Error Loss			
Convolution 1-D	0.064	0.987	0.063	0.991
LSTM	0.092	0.973	0.055	0.993
Parallel Convolution 1-D	**0.039**	**0.995**	**0.042**	**0.996**
Timedistributed LSTM	0.067	0.986	0.072	0.992
Bidirectional LSTM	0.085	0.976	0.063	0.991
	Cosine Loss			
Convolution 1-D	0.049	0.992	0.065	0.992
LSTM	0.098	0.973	0.16	0.943
Parallel Convolution 1-D	**0.047**	**0.993**	**0.041**	**0.997**
Timedistributed LSTM	0.073	0.985	0.073	0.992
Bidirectional LSTM	0.097	0.972	0.074	0.989
	Log Cosh Loss			
Convolution 1-D	0.053	0.991	0.052	0.994
LSTM	0.081	0.973	0.124	0.951
Parallel Convolution 1-D	**0.053**	**0.990**	**0.042**	**0.996**
Timedistributed LSTM	0.064	0.986	0.055	0.994
Bidirectional LSTM	0.093	0.971	0.064	0.991
	Cross Entropy Loss			
Convolution 1-D	0.059	0.989	0.056	0.992
LSTM	0.089	0.973	0.113	0.960
Parallel Convolution 1-D	**0.049**	**0.992**	**0.028**	**0.998**
Timedistributed LSTM	0.061	0.987	0.117	0.978
Bidirectional LSTM	0.085	0.975	0.080	0.988

amongst the implemented models. This represents that it is necessary to select suitable deep learning model for estimation of abundance fractions of endmembers. For JasperRidge dataset, parallel convolution 1-D model with mean square error loss function has been performing well. While parallel convolution 1-D model with cross entropy loss function has been performing well for Samson dataset. It is mentioned that JasperRidge dataset has four endmembers (Tree, Soil, Water, and Road) while Samson dataset has three endmembers (Soil, Tree, and Water). It is observed that the performance of loss functions may vary by using different number of endmembers. Hence, one possible reason for varying performance of the loss function is that the number of endmembers. Further, one may test the performance by considering different combination of endmembers and loss functions using different types of mixed datasets in order to find out other responsible factors. Furthermore, this study may be used for various applications like quantitative change detection.

REFERENCES

[1] V. Lodhi, D. Chakravarty, and P. Mitra, "Hyperspectral imaging for earth observation: Platforms and instruments," *Journal of the Indian Institute of Science*, vol. 98, no. 4, pp. 429–443, 2018.

[2] ——, "Hyperspectral imaging system: Development aspects and recent trends," *Sensing and Imaging*, vol. 20, no. 1, pp. 1–24, 2019.

[3] M. B. Stuart, A. J. McGonigle, and J. R. Willmott, "Hyperspectral imaging in environmental monitoring: a review of recent developments and technological advances in compact field deployable systems," *Sensors*, vol. 19, no. 14, p. 3071, 2019.

[4] R. Fraser, "Hyperspectral remote sensing of turbidity and chlorophyll a among nebraska sand hills lakes," *International journal of remote sensing*, vol. 19, no. 8, pp. 1579–1589, 1998.

[5] M. Teke, H. S. Deveci, O. Haliloğlu, S. Z. Gürbüz, and U. Sakarya, "A short survey of hyperspectral remote sensing applications in agriculture," in *2013 6th International Conference on Recent Advances in Space Technologies (RAST)*. IEEE, 2013, pp. 171–176.

[6] J. B. Collins and C. E. Woodcock, "An assessment of several linear change detection techniques for mapping forest mortality using multitemporal landsat tm data," *Remote sensing of Environment*, vol. 56, no. 1, pp. 66–77, 1996.

[7] L. Zhizhong, Y. Rihong, D. Fuxing, D. Peijun, Z. Xianfeng, T. Bingxiang, Z. Huijie, and S. Hongjun, "A review on the geological applications of hyperspectral remote sensing technology," in *2012 4th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*. IEEE, 2012, pp. 1–4.

[8] D. C. Heinz *et al.*, "Fully constrained least squares linear spectral mixture analysis method for material quantification in hyperspectral imagery," *IEEE transactions on geoscience and remote sensing*, vol. 39, no. 3, pp. 529–545, 2001.

[9] J. M. Nascimento and J. M. Dias, "Vertex component analysis: A fast algorithm to unmix hyperspectral data," *IEEE transactions on Geoscience and Remote Sensing*, vol. 43, no. 4, pp. 898–910, 2005.

[10] M. E. Winter, "N-findr: An algorithm for fast autonomous spectral endmember determination in hyperspectral data," in *Imaging Spectrometry V*, vol. 3753. International Society for Optics and Photonics, 1999, pp. 266–276.

[11] J. M. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot, "Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches," *IEEE journal of selected topics in applied earth observations and remote sensing*, vol. 5, no. 2, pp. 354–379, 2012.

[12] X. Xu and Z. Shi, "Multi-objective based spectral unmixing for hyperspectral images," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 124, pp. 54–69, 2017.

[13] W. Fan, B. Hu, J. Miller, and M. Li, "Comparative study between a new nonlinear model and common linear model for analysing laboratory simulated-forest hyperspectral data," *International Journal of Remote Sensing*, vol. 30, no. 11, pp. 2951–2962, 2009.

[14] Q. Qu, N. M. Nasrabadi, and T. D. Tran, "Abundance estimation for bilinear mixture models via joint sparse and low-rank representation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 7, pp. 4404–4423, 2014.

[15] Y. Altmann, A. Halimi, N. Dobigeon, and J.-Y. Tourneret, "Supervised nonlinear spectral unmixing using a postnonlinear mixing model for hyperspectral imagery," *IEEE Transactions on Image Processing*, vol. 21, no. 6, pp. 3017–3025, 2012.

[16] R. Heylen and P. Scheunders, "A multilinear mixing model for nonlinear spectral unmixing," *IEEE transactions on geoscience and remote sensing*, vol. 54, no. 1, pp. 240–251, 2016.

[17] J. F. Mas and J. J. Flores, "The application of artificial neural networks to the analysis of remotely sensed data," *International Journal of Remote Sensing*, vol. 29, no. 3, pp. 617–663, 2008.

[18] J. Benediktsson and J. Sveinsson, "Feature extraction for multisource data classification with artificial neural networks," *International Journal of Remote Sensing*, vol. 18, no. 4, pp. 727–740, 1997.

[19] J. Li, X. Li, B. Huang, and L. Zhao, "Hopfield neural network approach for supervised nonlinear spectral unmixing," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 7, pp. 1002–1006, 2016.

[20] X. Xu, Z. Shi, and B. Pan, "A supervised abundance estimation method for hyperspectral unmixing," *Remote Sensing Letters*, vol. 9, no. 4, pp. 383–392, 2018.

[21] B. Palsson, J. Sigurdsson, J. R. Sveinsson, and M. O. Ulfarsson, "Hyperspectral unmixing using a neural network autoencoder," *IEEE Access*, vol. 6, pp. 25 646–25 656, 2018.

[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[23] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio, "Object recognition with gradient-based learning," in *Shape, contour and grouping in computer vision*. Springer, 1999, pp. 319–345.

[24] F. Zhu, Y. Wang, S. Xiang, B. Fan, and C. Pan, "Structured sparse method for hyperspectral unmixing," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 88, pp. 101–118, 2014.

[25] F. Zhu, Y. Wang, B. Fan, G. Meng, S. Xiang, and C. Pan, "Spectral unmixing via data-guided sparsity," *CoRR*, vol. abs/1403.3155, 2014.

[26] F. Zhu, Y. Wang, B. Fan, G. Meng, and C. Pan, "Effective spectral unmixing via robust representation and learning-based sparsity," *CoRR*, vol. abs/1409.0685, 2014.