Multispectral Image Segmentation With Dimensionality Reduction Using Autoencoders

Eliton Albuquerque and Claudio R. Jung
Institute of Informatics, Federal University of Rio Grande do Sul
Emails: {jeafilho,crjung}@inf.ufrgs.br

***Abstract*—Autoencoder (AE) implementations through neural networks have achieved impressive results on dimensionality reduction tasks, such as multispectral (MS) imagery compression. Dimensionality reduction algorithms are necessary when dealing with large multispectral datasets, since the data captured by multiple levels of narrow spectral wavelengths causes high processing and storage costs, particularly when such highly dimensional MS data are used as input to deep learning networks. Traditional data compression techniques like Principal Component Analysis (PCA) are popular in remote sensing applications. However, its implementation on MS data may make the data unusable for computer vision (CV) tasks such as image segmentation, especially when applying severe compression. On the other hand, AEs provide great generalization capabilities over complex data, especially when combined with other CV pipelines. For the relevant problem of semantic segmentation, the results are considerably degraded when using dimensionality-reduced images with PCA. When using vanilla autoencoders trained with the traditional MSE loss, the segmentation results improve over PCA but are still considerably behind the one obtained with uncompressed data, which indicates a potential domain shift. In this work, we show that training an AE using a combination of the MSE loss and an additional proxy loss based on a pre-trained segmentation module can significantly improve the AE restoration process, alleviating the accuracy drop of semantic segmentation even for strong compression rates. Our code is available at https://github.com/elitonfilho/pca.**

I. Introduction

The generation of massive and complex databases that holds high-dimensional data could be useful in the recent era of big data, especially when fuelled by multiple data engineering methods that aim to extract the greatest number of features for the same observation. Such massive data structures hold information that is not used effectively most of the times, bringing up the classical "curse of dimensionality" problem [1]–[3]: high-dimensional data deteriorates the performance of conventional learning methods. Besides that, high-dimensional data induces increased complexity, higher computational burden, harder interpretability, and higher risk of overfitting.

Advances in imaging devices and high-throughput data collection technologies have led to a large data capture in various domains. Amongst them, multispectral (MS) imagery is well known for generating great volume of data due to an elevated number of spectral bands, high spatial resolution, or high capture rate [4], which is usually the case for current imaging solutions such as satellites or drone mapping. The use of multispectral images is reported in a variety of applications such as agronomy [5], [6] and surveillance [7].

In this context, data compression techniques might be useful for reducing the amount of stored data without significant loss of information. When the number of spectral bands is large (as in most hyperspectral sensors), dimensionality reduction might even improve some tasks, such as classification/segmentation, particularly when the number of training data is small (Hughes phenomenon [1]).

Principal Component Analysis (PCA) is the most popular technique for dimensionality reduction [8], in particular for remotely sensed data [9]. PCA uses orthogonal transformations to obtain a set of linear uncorrelated variables (principal components), being extended in multiple approaches such as Kernel PCA [10] and Sparse PCA [11], which allows dealing with non-linearities. The development of efficient PCA algorithms leveraged the pre-processing of MS image datasets to eliminate data redundancy while keeping most of the relevant information [12]. Using PCA-reduction in deep learning applications has a noticeable good point: generating a smaller dataset while maintaining a high correlation to the original information, which allows for accelerating the training and inference processes, since more images can be allocated at once in the memory. However, dimensionality reduction might destroy discriminative information, which is relevant for classification tasks.

Autoencoders (AEs) were already studied in the '90s as an alternative to dimensionality reduction techniques, but practical results just appeared in the last decade, when we overcame the difficulty in training complex multi-layer neural networks [13]. Multiple studies [14], [15], including research involving MS imagery data [16], [17], showed that autoencoders have great potential for learning intrinsic properties of high-dimensional data. Amongst AE variations, one of the most common is the Variational autoencoder [18], whose training is regularized by encoding the inputs as a distribution over the latent space.

Regardless of the chosen dimensionality reduction approach, the reconstructed signal usually is not error-free (particularly for higher compression rates). Hence, when restoring compressed images, the reconstruction error might be propagated to other tasks relevant to MS imaging, such as semantic segmentation. This problem might be particularly relevant when using deep learning approaches, since a possible domain shift between the reconstructed and the original images might considerably degrade the inference results of neural networks.

In this work, we initially evaluate the impact of different

data dimensionality reduction approaches in semantic segmentation tasks. More precisely, we first perform dimensionality reduction using PCA or vanilla AEs trained with the traditional Mean Squared Error (MSE) loss. We then fed the reconstructed images to a pre-trained segmentation module, and noted that results with the vanilla AE are substantially better than PCA considering the segmentation task for larger compression rates. However, they are still considerably worse compared to using uncompressed data, which indicates a potential domain shift in the reconstruction process. To mitigate this problem, we explore a combination of the MSE loss with a task-driven loss – which is a segmentation error metric in our case – to train an AE. We use a simple ResNet module [19] that employs residual learning and identity mapping shortcuts as the chosen baseline AE (we also tested a U-Net [20], but the results were inferior). Our results indicate that adding the segmentation proxy to train the AE provides an accuracy boost to the segmentation results, even when reducing the dimensionality to a single channel.

The rest of the paper is organized as follows. In Section III, we propose an autoencoder architecture as a framework for dimensionality reduction while detailing our experimentation procedure. In Section IV, we discuss the proposed method and illustrate the experimental results on AE/PCA reconstructed data, finally concluding the paper in Section V.

II. Related work

While early literature on Principal Component Analysis dates from the early 20th century [21], it was only with recent electronic development that its implementation was computationally feasible on large data structures. It is employed in multiple computer vision tasks, either as a core idea in deep learning architectures or as a simple pre-processing tool [22]. This very well-known statistical technique is quite common when processing raster data: multiple measurements of raster surfaces can aggregate an exorbitant volume of observations, especially when dealing with multispectral data, and Principal Component Analysis is a great approach to lower dimensional spaces while maintaining spatial correlation between variables [23]. Recent extensions of PCA, such as robust PCA [24], bring up important methodological advances to tackle very large datasets, which is often the case in image processing and machine learning applications.

Recent hardware developments also enabled the use of complex autoencoders to capture multispectral data distributions. autoencoders are mostly employed on image restoration tasks. Amongst them, convolutional architectures are quite common in image fusion [25] and unmixing tasks [26]. There are also some authors that explore image compression on encoder-decoder compositions, since its forward pass until a latent space can simulate data compression. This is the case of Kong et al. [27], who employ a spatial attention network based on variational autoencoders that offer state-of-the-art compression on 7-band Landsat-8 and 8-band WorldView-3 imagery. Alves et al. [28] explore a joint compression and denoising method using Convolutional autoencoder (CAE) and generalized divisive normalization. Cheng et al. [29] also use a convolutional autoencoder to propose a lossy image compression method feeding PCA-rotated feature maps generated by the CAE into a quantization equation.

After the consolidation of task-tailored neural networks capable of providing state-of-the-art solutions for most computer vision challenges, the aggregation of multi-modal deep learning tasks became increasingly common. The usage of domain-specific information contained in the training signals of related tasks has a clear advantage: the potential for improved performance if the associated tasks share complementary information, or the inter-regularization of tasks, without necessarily improving much of the resources footprint due to their inherent layer sharing [30]. Some applications of multispectral imagery are proposed by Feng et al. [31] by mixing super-resolution, colorization, and pan-sharpening to produce realistic-looking images using only panchromatic images as inputs, and He et al. [32] that submitted a multi-object tracking method trained on satellite images that uses a graph-based spatiotemporal module and a multitask gradient adversarial learning strategy. Even though there is a vast literature about task-driven deep learning methods, the impacts of data compression on task-driven models remain a relatively nascent subject.

III. The Proposed Approach

Our main goal in this work is to propose a dimensionality reduction approach using an AE trained with a loss function that combines reconstruction and segmentation error terms. The input of the AE is a $c_i \times h \times w$ multispectral image, and the output is $c_b \times h \times w$. Here, $h \times w$ is the spatial resolution of the image, and $c_i > c_b$ are the number of channels of the original and dimensionality-reduced images, respectively.

The encoder module of the proposed AE consists of a set of convolutional blocks with (stride, padding, kernel size) = $(1, 1, 3)$, and the number of output channels decreases by a factor of 2 for every consecutive block after the first block, which has 64 channels. The last block of the encoder has $c_b < c_i$ channels, and its output will produce the dimensionality-reduced version of the input image. Since we want to reduce storage requirements, it is typically desired to have $c_b << c_i$, with the extreme case being $c_b = 1$. The decoder has a similar architecture with blocks used in a reversed order: it employs the same type of convolutional blocks to expand the depth of the bottleneck's feature maps from c_b to c_o. In our experiments, we adopt $c_i = c_o$ to simulate the autoencoder task. The proposed architecture is shown in Fig. 1. Notice that, while the amount of decoder blocks is constant, the quantity of residual encoder blocks depends on n_o, being equal to $\log_2(64/c_b) + 1$.

We chose to use a simple and efficient ResNet [19] module as the segmentation network, since its deep representation presents a good generalization performance on pattern recognition tasks. More precisely, we chose the ResNet-101 variant, which employs a stack of 101 layers with residual connections

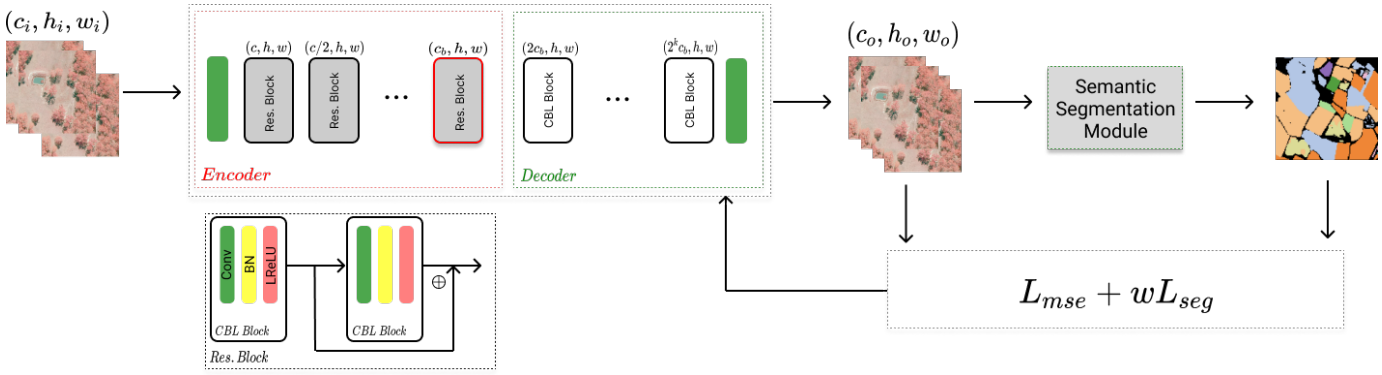


Fig. 1. Representation of the employed autoencoder architecture. The blocks are composed of convolutions (green), batch normalizations (yellow), and Leaky-ReLU activations (red). In the bottom left, we show the representation of the residual architecture employed in the autoencoder module. The outlined red block at the end of the encoder sub-module represents the bottleneck of the AE network.

in-between, and trained it using a cross-entropy residual loss between the predicted and original masks. The segmentation module is trained on uncompressed data (c_i channels), generating a pre-trained network that will be used both as an evaluation oracle and a segmentation proxy for the AE.

The dimensionality-reduction AE is trained using two different strategies: the first one uses only the MSE between the original image x and the output of the decoder $\hat{x}$ – the reconstruction loss, which is the usual approach for training AEs. The second approach explores a joint loss function that combines the reconstruction and segmentation error terms. This latter error term involves a set of per-pixel pseudo-labels produced by the segmentation proxy, encoded as one-hot vectors $\boldsymbol{l} = [l_k]$, and the corresponding outputs produced by the segmentation module $\hat{\boldsymbol{l}} = [\hat{l}_k]$, coming from a softmax layer. The joint loss is given by

$$\mathcal{L} = \mathcal{L}_{mse} + \omega \mathcal{L}_{seg}, \tag{1}$$

where

$$\mathcal{L}_{mse} = \frac{1}{hwc_i} \sum_{c=1}^{c_i} \sum_{i=1}^{h} \sum_{j=1}^{w} \left(x(c,i,j) - \hat{x}(c,i,j)\right)^2 \tag{2}$$

is the classical MSE error computed across all channels between the uncompressed image and the prediction produced by the decoder, and

$$\mathcal{L}_{seg} = \frac{1}{hw} \sum_{i=1}^{h} \sum_{j=1}^{w} \sum_{k=1}^{C} -l_k(i,j) \log(\hat{l}_k(i,j)) \tag{3}$$

is the average per-image cross-entropy loss, which aims to enforce the consistency of the semantic pseudo-labels produced by the original high-dimensional data and its compressed version. Parameter C denotes the number of classes, and ω is the weight (set to 0.01 based on experiments) that balances the magnitude of both loss terms.

To train both the segmentation module and the AE, we used the Adam [33] optimizer with decay $(\alpha, \beta) = (0.9, 0.999)$. The training process is conducted for 200 epochs for each network, and the learning rate follows a multi-step update, being set to $(1^{-3}, 5^{-4}, 1^{-4})$ on $(50^{th}, 100^{th}, 150^{th})$ epochs, respectively. We used PyTorch 1.10, supported by Python 3.8.10 and CUDA 11.3 to train and validate our models.

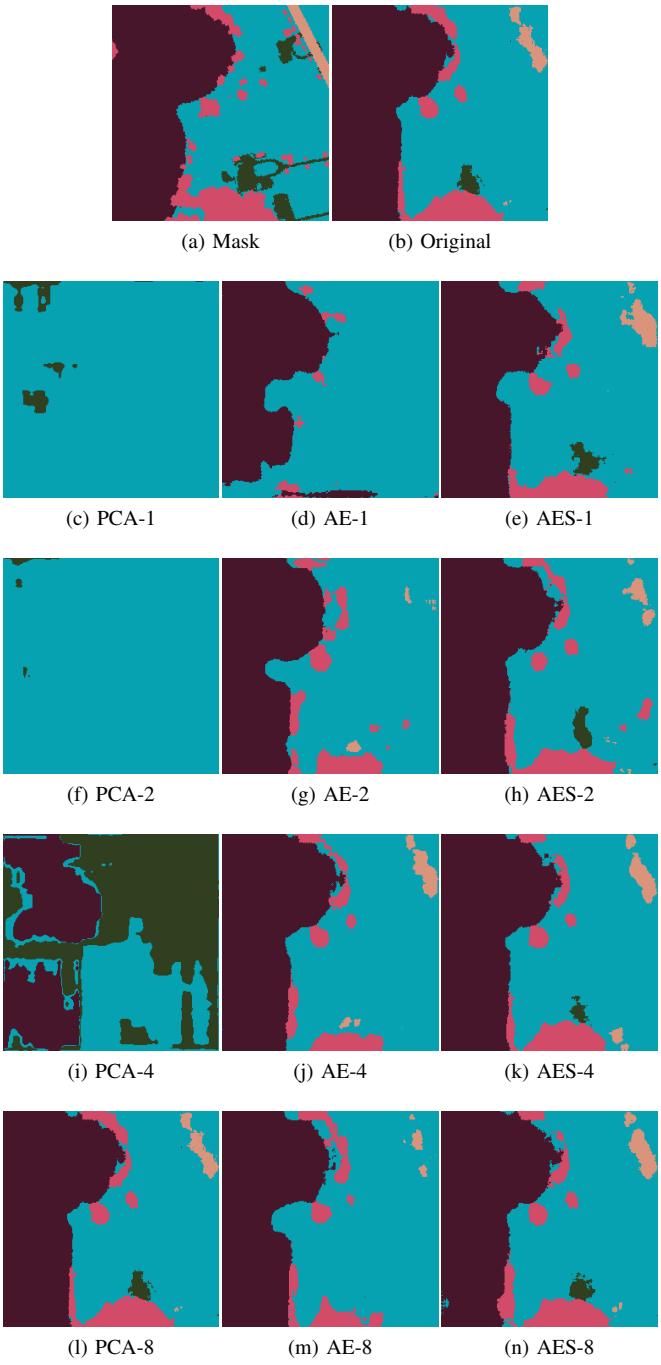
IV. Experimental Results

Since our main evaluation task is semantic segmentation, we need a multispectral dataset with class labels. We used the Chesapeake Land Cover dataset [34], which contains land cover labels for six classes (water, tree canopy/forest, low vegetation/field, barren land, roads, and other impervious surfaces) and Landsat8 surface reflectance imagery for nine bands (i.e., $c_i = 9$). We used the tiles from the region of Delaware, normalizing the values of the multispectral bands to the range $[0, 1]$ while keeping other characteristics devised by the dataset creators, such as the patch size (256×256) and the train/val split. Since the AE is fully convolutional, the whole test images are fed to the network in the inference phase.

The segmentation network is trained only once using the uncompressed (9-channel) images, and we trained eight different AEs: The first four were obtained using only the MSE loss (reconstruction) varying the number of channels $c_b \in \{1, 2, 4, 8\}$ in the encoder bottleneck, and they are denoted as AE-1, AE-2, AE-4, and AE-8. The other four AEs were trained using the joint loss function with the segmentation term provided in Eq. (1), named AES-1, AES-2, AES-3, AES-4.

To validate the quality of the proposed AEs, we computed segmentation figures of merit by applying the pre-trained segmentation network to the decoded (de-compressed) versions of the test images varying the number of channels (c_b) of the corresponding encoder. We also compared our results with PCA-encoded versions of the multispectral images using a varying number of features (PCA-1, PCA-2, PCA-4, and PCA-8), which are decoded and fed to the same segmentation network.

Table I shows the segmentation results using different feature reduction techniques (PCA, AE, and AES) varying the target number of features $c_b \in \{1, 2, 4, 8\}$. The best result for each value of c_b is shown in bold, and "Original" relates



(a) Mask (b) Original

(c) PCA-1 (d) AE-1 (e) AES-1

(f) PCA-2 (g) AE-2 (h) AES-2

(i) PCA-4 (j) AE-4 (k) AES-4

(l) PCA-8 (m) AE-8 (n) AES-8

Fig. 2. Segmentation results for the PCA, AE and AES experiments with bottleneck 1, 2, 4 and 8. The class colors are violet (water), forest (pink), roads (gray), impervious surface (green) and field (blue)

to segmentation results using the uncompressed dataset. We used the accuracy, mean accuracy, class-wise intersection over union (IoU), and the overall mean value (mIoU) between the output of the segmentation module and the ground truth mask as figures of merit, with the IoU class mappings following the order: 1– water, 2– forest, 3– field / low vegetation, 4– barren land, 5– roads and 6– other impervious surfaces. We additionally show the MSE between the reconstructed and the original images in the last column of Table I.

Table I indicates that PCA compression using medium to severe bottlenecks $c_b \in \{1, 2, 4\}$ yields very poor segmentation results, but very light dimensionality reduction ($c_b = 8$) showed virtually the same segmentation performance of the uncompressed dataset (i.e., using the full nine channels). In fact, the cumulative variance using eight (out of nine) bands of the PCA reconstructed dataset yields more than 99% of the entire dataset variance, allowing a faithful representation of the original dataset (which is corroborated by a very small reconstruction MSE). The AE approach showed a strong accuracy gain on the segmentation task compared to PCA with the same number of features, especially on small bottlenecks (large compression ratios). We notice, however, a considerable IoU drop in some classes (particularly 4, 5, and 6, which represent the impervious surfaces and barren land), which we believe was caused by a domain shift between the decoded version and the original image for the task of segmentation.

The AE version with the segmentation proxy (AES) presented considerably better results, particularly for very small bottlenecks. For example, AES-1 (reduction to a single channel) presents larger mIoU values than AE-4 and much larger than PCA-4, which both encode the compressed versions with $4\times$ more channels (it is even slightly better than AE-8). Compared to the original images, AES-1 achieves 86.7% of the segmentation mIoU with a channel-wise compression factor of $9\times$. We believe that using the segmentation module in the joint training process alleviates the domain shift problem, since the segmentation module regularizes the AE to reconstruct images that are more loyal to the class predictions. Using this relatively simple approach introduces a slight overhead in the training process (the use of the segmentation proxy, which presents fixed weights that are not updated) and has no impact on the architecture of the AE.

It is also interesting to note in Table I that AES provides reconstructed images with smaller MSE than AE for $c_b \in \{2, 4, 8\}$, even though AE is trained only with MSE as the loss. We believe that using the segmentation proxy might also work as a regularizer for the autoencoder, leading to smaller MSE on unseen data.

We also show in Fig. 2 the segmentation results produced by PCA, AE, and AES dimensionally-reduced data (with different compression factors). We note that reconstructed images with PCA yield very poor segmentation results for stronger compression rates (most of the image was classified as "field" for $c_b \in \{1, 2\}$, and large portions of "impervious surface" were wrongly detected with $c_b = 4$). Results using AE are able to retrieve the most predominant classes (such as water, forest, and fields) even for large compression ratios, but fail to capture detailed and less frequent classes. This problem is alleviated with the introduction of the joint learning approach, and even AES-1 is able to identify less frequent categories such as road and impervious land classes. It is also interesting to note that results for AES using $c_b = 1$ and $c_b = 8$ are visually similar, which indicates that AES-1 might be a very good choice if large compression rates are desired.

TABLE I
EVALUATION OF METRICS GENERATED BY THE SEGMENTATION MODULE ON ORIGINAL OR RECONSTRUCTED DATA

Experiment	Acc	mAcc	IoU	mIoU	MSE
PCA-1	0.063	0.167	{1: 0.0, 2: 0.0, 3: 0.07, 4: 0.0, 5: 0.016, 6: 0.0}	0.014	$1.04e^{-1}$
AE-1	0.886	0.422	{1: 0.335, 2: 0.742, 3: 0.866, 4: 0.0, 5: 0.012, 6: 0.029}	0.331	$2.32e^{-3}$
AES-1	0.930	0.558	{1: 0.834, 2: 0.86, 3: 0.913, 4: 0.042, 5: 0.164, 6: 0.278}	**0.515**	$2.59e^{-3}$
PCA-2	0.423	0.191	{1: 0.0, 2: 0.002, 3: 0.512, 4: 0.001, 5: 0.019, 6: 0.003}	0.089	$8.06e^{-2}$
AE-2	0.854	0.450	{1: 0.735, 2: 0.628, 3: 0.823, 4: 0.0, 5: 0.107, 6: 0.118}	0.402	$2.14e^{-3}$
AES-2	0.933	0.578	{1: 0.858, 2: 0.866, 3: 0.916, 4: 0.085, 5: 0.19, 6: 0.299}	**0.536**	$8.70e^{-4}$
PCA-4	0.248	0.161	{1: 0.057, 2: 0.001, 3: 0.269, 4: 0.0, 5: 0.016, 6: 0.01}	0.059	$3.85e^{-2}$
AE-4	0.923	0.501	{1: 0.81, 2: 0.842, 3: 0.904, 4: 0.0, 5: 0.099, 6: 0.162}	0.469	$7.00e^{-4}$
AES-4	0.938	0.612	{1: 0.879, 2: 0.876, 3: 0.921, 4: 0.168, 5: 0.201, 6: 0.357}	**0.567**	$3.70e^{-4}$
PCA-8	0.944	0.638	{1: 0.894, 2: 0.89, 3: 0.928, 4: 0.248, 5: 0.21, 6: 0.394}	**0.594**	$8.00e^{-7}$
AE-8	0.928	0.545	{1: 0.851, 2: 0.851, 3: 0.908, 4: 0.058, 5: 0.153, 6: 0.249}	0.512	$3.80e^{-4}$
AES-8	0.937	0.610	{1: 0.866, 2: 0.872, 3: 0.92, 4: 0.159, 5: 0.203, 6: 0.374}	0.566	$3.00e^{-4}$
Original	0.944	0.638	{1: 0.894, 2: 0.89, 3: 0.928, 4: 0.248, 5: 0.21, 6: 0.394}	**0.594**	-

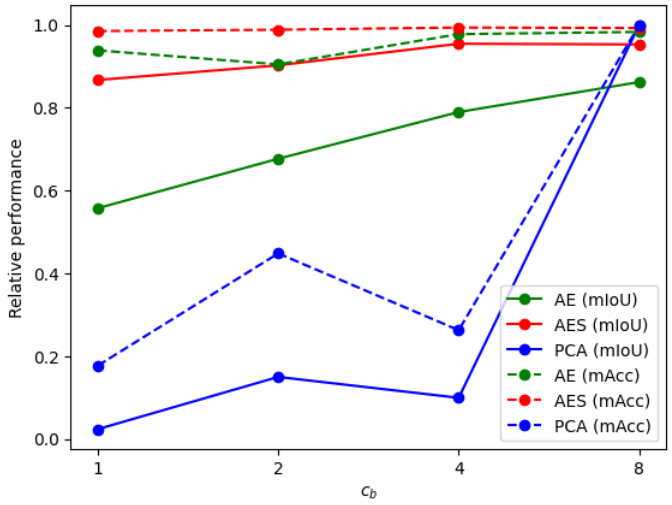


Fig. 3. Relative performance for the semantic segmentation task between experiments PCA-n, AE-n, and AES-n ($n \in \{1, 2, 4, 8\}$) and the original 9-band input. The continuous line displays mean IoU values, while dotted lines show the mean accuracy.

Finally, Fig. 3 summarizes the relative performance in the semantic segmentation task of reconstructed inputs from the PCA, AE, or AES methods and the original 9-band input. It shows the power of the task-oriented autoencoder in high-compressed multispectral data: AES-1 can reach obtain almost 87% of mIoU of the core experiment with non-compressed data while using only 11% of the original image size (channel-wise reduction). Besides, the storage requirements for the model should be mostly negligible when compared to large multispectral datasets. On the other hand, both PCA and AE present a more consistent accuracy drop when we decrease b.

V. CONCLUSIONS

This paper presented a task-driven dimensionality reduction technique based on deep autoencoders. The core idea of the paper is that by using an error metric of the main task as a proxy to guide the training process of the autoencoder can regularize the reconstruction process through the decoder when compared to the traditional reconstruction loss only.

Our results indicate that standard approaches such as PCA are adequate for very light reductions (which are not useful in practical applications, since they yield very low compression rates), but tend to produce very poor segmentation results on mild and severe dimensionality reductions. On the other hand, autoencoders showed to be a viable solution for strong reductions: results using only the reconstruction MSE as the loss (named AE-c_b) are considerably better than PCA for mild to extreme compression factors. The full approach with the segmentation proxy (AES-c_b) improved over AE-c_b for all tested compression factors, and even dimensionality reduction to a single feature ($c_b = 1$) can lead to interesting segmentation results: approximately 86.7% of the mIoU obtained with the uncompressed raw data, but using only 11.1% of the channels.

We believe this work could serve as a starting point for deeper research involving multiple sources of MS data, state-of-art autoencoders, and segmentation models. Although we used semantic segmentation as the driving task, the same strategy can be used for other tasks as well, such as object detection. In future work, we intend to work with hyper-spectral data as well, for which the proposed dimensionality reduction approach might even improve segmentation results over uncompressed data (Hughes phenomenon [1]). We also plan to investigate different AE architectures, and evaluate the generalization capabilities in cross-dataset experiments.

ACKNOWLEDGMENT

This study was financed in part by CNPq and Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

REFERENCES

[1] G. Hughes, "On the mean accuracy of statistical pattern recognizers," *IEEE transactions on information theory*, vol. 14, no. 1, pp. 55–63, 1968.

[2] W. Jia, M. Sun, J. Lian, and S. Hou, "Feature dimensionality reduction: a review," *Complex & Intelligent Systems*, pp. 1–31, 2022.

[3] X. Huang, L. Wu, and Y. Ye, "A review on dimensionality reduction techniques," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 33, no. 10, p. 1950017, 2019.

[4] V. D. Vaughn and T. S. Wilkinson, "System considerations for multispectral image compression designs," *IEEE Signal Processing Magazine*, vol. 12, no. 1, pp. 19–31, 1995.

[5] J. J. Assmann, J. T. Kerby, A. M. Cunliffe, and I. H. Myers-Smith, "Vegetation monitoring using multispectral sensors—best practices and lessons learned from high latitudes," *Journal of Unmanned Vehicle Systems*, vol. 7, no. 1, pp. 54–75, 2018.

[6] C. Tang, H. He, E. Li, and H. Li, "Multispectral imaging for predicting sugar content of 'fuji'apples," *Optics & Laser Technology*, vol. 106, pp. 280–285, 2018.

[7] J. Liu, S. Zhang, S. Wang, and D. N. Metaxas, "Multispectral deep neural networks for pedestrian detection," *arXiv preprint arXiv:1611.02644*, 2016.

[8] S. Velliangiri, S. Alagumuthukrishnan *et al.*, "A review of dimensionality reduction techniques for efficient computation," *Procedia Computer Science*, vol. 165, pp. 104–111, 2019.

[9] J. A. Richards and J. Richards, *Remote sensing digital image analysis*. Springer, 1999, vol. 3.

[10] S. Mika, B. Schölkopf, A. Smola, K.-R. Müller, M. Scholz, and G. Rätsch, "Kernel pca and de-noising in feature spaces," *Advances in neural information processing systems*, vol. 11, 1998.

[11] C.-M. Feng, Y.-L. Gao, J.-X. Liu, C.-H. Zheng, S.-J. Li, and D. Wang, "A simple review of sparse principal components analysis," in *International Conference on Intelligent Computing*. Springer, 2016, pp. 374–383.

[12] A. Agarwal, T. El-Ghazawi, H. El-Askary, and J. Le-Moigne, "Efficient hierarchical-pca dimension reduction for hyperspectral imagery," in *2007 IEEE International Symposium on Signal Processing and Information Technology*. IEEE, 2007, pp. 353–356.

[13] Y. Wang, H. Yao, and S. Zhao, "Auto-encoder based dimensionality reduction," *Neurocomputing*, vol. 184, pp. 232–242, 2016.

[14] Q. V. Le, "Building high-level features using large scale unsupervised learning," in *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, 2013, pp. 8595–8598.

[15] Z.-h. Hu and Y.-l. Song, "Dimensionality reduction and reconstruction of data based on autoencoder network," , vol. 31, no. 5, pp. 1189–1192, 2009.

[16] 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.

[17] M. Ramamurthy, Y. H. Robinson, S. Vimal, and A. Suresh, "Auto encoder based dimensionality reduction and classification using convolutional neural networks for hyperspectral images," *Microprocessors and Microsystems*, vol. 79, p. 103280, 2020.

[18] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[20] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[21] H. Hotelling, "Analysis of a complex of statistical variables into principal components." *Journal of educational psychology*, vol. 24, no. 6, p. 417, 1933.

[22] C. Witharana, M. A. E. Bhuiyan, A. K. Liljedahl, M. Kanevskiy, H. E. Epstein, B. M. Jones, R. Daanen, C. G. Griffin, K. Kent, and M. K. W. Jones, "Understanding the synergies of deep learning and data fusion of multispectral and panchromatic high resolution commercial satellite imagery for automated ice-wedge polygon detection," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 170, pp. 174–191, 2020.

[23] U. Demšar, P. Harris, C. Brunsdon, A. S. Fotheringham, and S. McLoone, "Principal component analysis on spatial data: an overview," *Annals of the Association of American Geographers*, vol. 103, no. 1, pp. 106–128, 2013.

[24] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM (JACM)*, vol. 58, no. 3, pp. 1–37, 2011.

[25] A. Azarang, H. E. Manoochehri, and N. Kehtarnavaz, "Convolutional autoencoder-based multispectral image fusion," *IEEE access*, vol. 7, pp. 35 673–35 683, 2019.

[26] J. S. Bhatt and M. V. Joshi, "Deep learning in hyperspectral unmixing: A review," in *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium*, 2020, pp. 2189–2192.

[27] F. Kong, T. Cao, Y. Li, D. Li, and K. Hu, "Multi-scale spatial-spectral attention network for multispectral image compression based on variational autoencoder," *Signal Processing*, vol. 198, p. 108589, 2022.

[28] V. Alves de Oliveira, M. Chabert, T. Oberlin, C. Poulliat, M. Bruno, C. Latry, M. Carlavan, S. Henrot, F. Falzon, and R. Camarero, "Reduced-complexity end-to-end variational autoencoder for on board satellite image compression," *Remote Sensing*, vol. 13, no. 3, p. 447, 2021.

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

[30] S. Vandenhende, S. Georgoulis, W. Van Gansbeke, M. Proesmans, D. Dai, and L. Van Gool, "Multi-task learning for dense prediction tasks: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 7, pp. 3614–3633, 2021.

[31] J. Feng, Q. Jiang, C.-H. Tseng, X. Jin, L. Liu, W. Zhou, and S. Yao, "A deep multitask convolutional neural network for remote sensing image super-resolution and colorization," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, 2022.

[32] Q. He, X. Sun, Z. Yan, B. Li, and K. Fu, "Multi-object tracking in satellite videos with graph-based multitask modeling," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–13, 2022.

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[34] C. Robinson, L. Hou, K. Malkin, R. Soobitsky, J. Czawlytko, B. Dilkina, and N. Jojic, "Large scale high-resolution land cover mapping with multi-resolution data," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 726–12 735.