Recovery of Damaged Text using Variational Auto Encoders: An Application Study

Brock Dyer
Department of Computer Science
Rochester Institute of Technology
Rochester, NY 14623
bld4524@rit.edu

Zoe LaLena
Department of Imaging Science
Rochester Institute of Technology
Rochester, NY 14623
zel9356@rit.edu

Dade Wood
Department of Computer Science
Rochester Institute of Technology
Rochester, NY 14623
daw1882@rit.edu

Abstract

Old parchment can contain a wealth of information besides that which is seen on the surface. Palimpsests are parchment documents which have been reused by either mechanically or chemically removing the original text in order to write new text on the surface. Additionally, reagents were often used in the 1800s in order to improve legibility of these parchments, however, over time the chemicals would permanently damage the documents, increasing the difficulty of recovering the text. Standard methods at recovering the original text involve using Principal Component Analysis (PCA) in order to reduce the dimensionality of the data and hopefully bring out important information. We propose to instead use Variational Auto Encoders (VAE) to learn a low dimensional embedding containing extracted text pixels from the reagent damaged palimpsests.

1 Introduction

In Rochester New York a cultural heritage imaging boom has occurred in the past couple of decades, with universities like Rochester Institute of Technology and the University of Rochester imaging and transcribing historical manuscripts that have been damaged. Recently there has been an increase in the interest of reagent treated parchment documents in particular.

In order to improve the legibility of metallic inks on parchment, chemicals called reagents were used. These chemicals would react in ways that made the metallic inks darker and easier to read. According to Albrecht there were three main substances used as reagents. The three substances are tincture of oak-gall, liver of sulfur tincture and Giobert tincture. Each of the reagents would succeed in improving legibility, but over time they would leave permanent blemishes in the parchments, and generally rendering the text illegible [1]. Since reagents were mostly used in the 1800s, the text that was damaged hundred of years ago is now lost to time. Figure 1 shows how destructive reagents can be. Reagents were commonly used on palimpsests, palimpsests are documents that had text removed so the parchment could be reused. The text would either be scraped away or chemically removed. Palimpsests were a great interest of scholars in the 19th century, and still are today. In order to improve the legibility of removed text, scholars applied reagents to palimpsest frequently. Due to the damage reagents cause, both the removed text and the over text on reagent treated palimpsest are illegible.

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

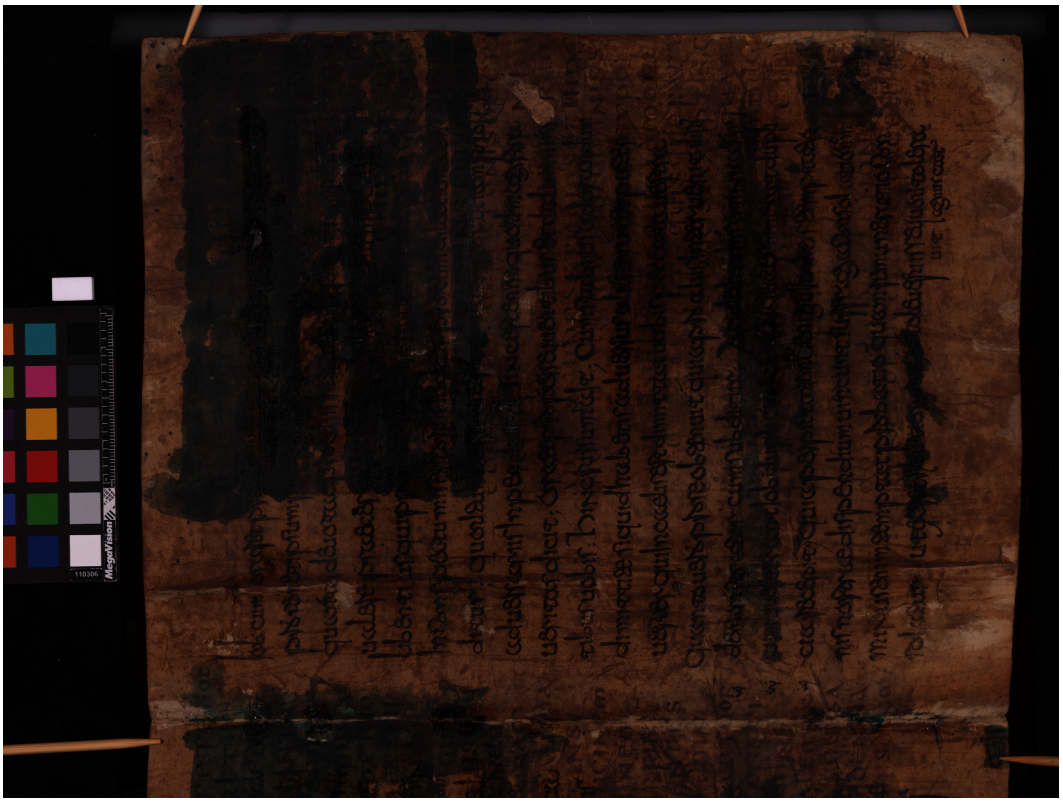

Figure 1: AN Example of a reagent treated manuscript. Reagent has been heavily applied to the left side of the manuscript, making the text in that location illegible. This document is Manuscript 38 from the Biblioteca Capitolare in Verona.

We propose an application study to determine if VAEs can be used to aid in the recovery of text obscured by reagents. Previous work has focused on finding the non-linear, and linear, lower dimension manifold that multi-spectral data of these reagent treated manuscripts lies on. Once the dimensionality of the data has been reduced, ideally the pixels that make up the text class can be easily found and extracted. They may naturally cluster together, or may be more easily found with machine learning techniques like Convolutional Neural Networks (CNN).

2 Related Work

Commonly when presented with multi spectral data of any manuscript, principal component analysis (PCA) will be run. PCA allows scientists and historians to find a a lower dimensional manifold that the original data lies on through a linear transformation. For reagent treated documents PCA does not work well. This is because the spectral qualities of the parchment, ink and reagent combine non linearly. So the principle components of reagent treated documents don't tent to reveal the text obscured by the reagent.

Previous work has been done to find lower dimensional representations of multi-spectral data of cultural artifacts. Cao was able to classify different pigment types using Laplacian Eigenmaps. Laplacian Eigenmaps allows one to easily find a non-linear manifold that the data lies on. Cao firsts converts the multi-spectral image data to graph using a density weighted k nearest neighbor algorithm where closeness is defined by spectral similarity. Using the weight (W) matrix from graph, the degree matrix can be found. The degree matrix is simply the degree (D) of of every node along the diagonal of the matrix and zeros elsewhere.

$$D_{ii} = \Sigma_{j=1}^{m} W_{ij} \tag{1}$$

With D and W the graph Laplacian can be found as

$$L = W - D \tag{2}$$

The graph Laplacian (L) is then used to solve the Eigenvalue problem

$$L\Phi = D\Phi\Lambda \tag{3}$$

where the first l eigenvectors, that have the smallest (but not zero) Eigenvalues, are the new basis vectors in Laplacian space [2]. This results in a lower dimensional representation of the original data. This representation allowed Cao to use a CNN to classify different types of inks on manuscripts.

An adaption of Cao's method is being worked on by LaLena in separate research. The method is slow and does not show promise for heavily reagent treated sections of document. The method would take over 20 days to run on the entire image of manuscript 38. An example of a preliminary result from LaLena's method on manuscript 38 is shown in figure 2. We propose using VAEs to speed up the process and find a better lower dimensional representation of the data.

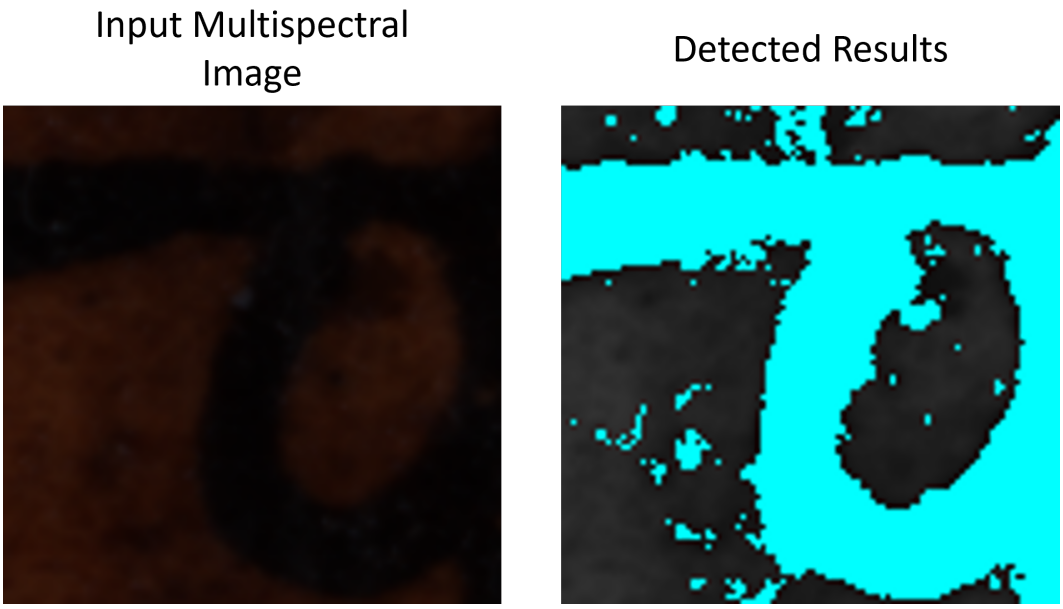

Figure 2: Detection of text on a subsection of manuscript 38 using the method proposed by LaLena.

3 Model Design

The proposed neural architecture that this approach uses is an application of the spectral-spatial variational autoencoder (U_{Hfe}SRVAE) introduced in Yu et. al. 2021. This model was chosen as it attempts to utilize information in hyper-spectral images to create a latent embedding distribution that can be used to classify pixel vectors as belonging to a specific class. The model works on each pixel vector of the image and generates a latent representation for that pixel vector. It also modifies the standard VAE approach with CNN and LSTM networks to capture spatial information around the pixel vector and then modifies the mean parameter of the VAE's latent representation with this spatial information. This approach is motivated by the idea that the context around a pixel vector is important to determining what that pixel vector represents. Modifying the mean parameter of the latent distribution with this spatial information should provide this additional information to an analysis task that uses the latent information.

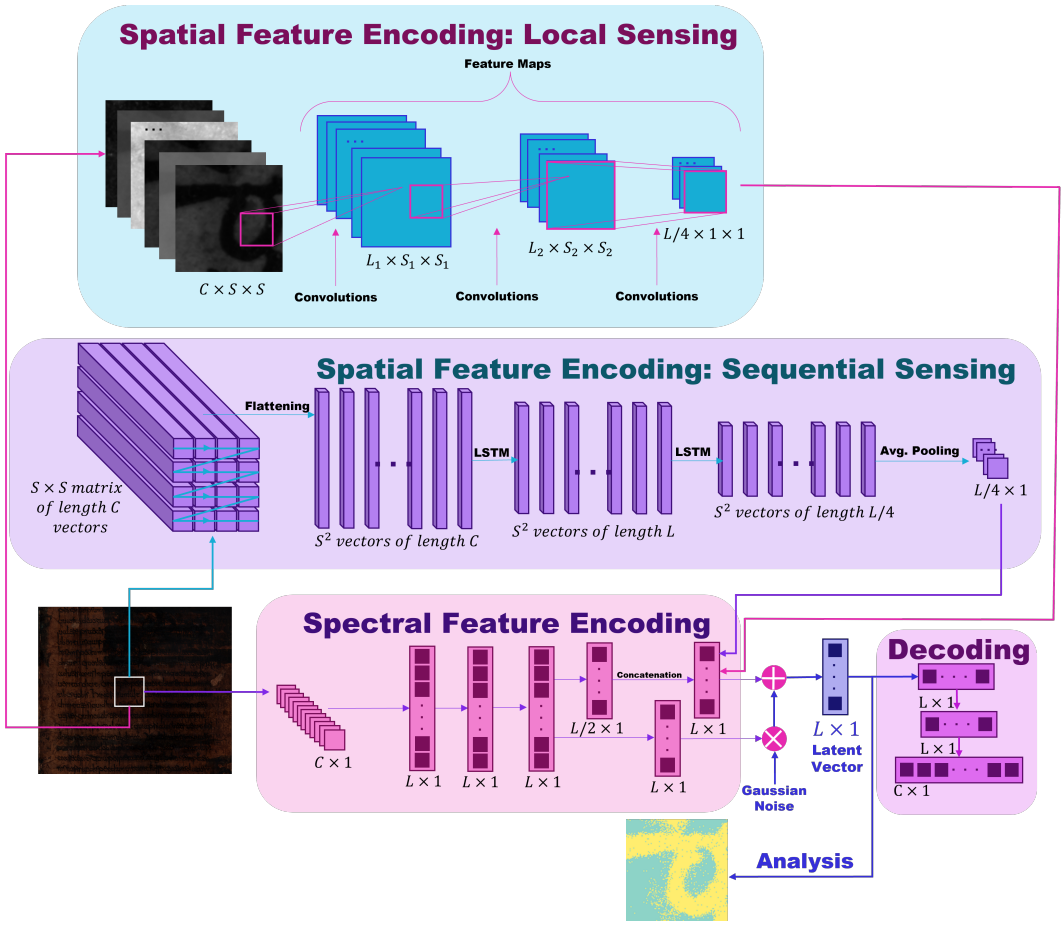

Figure 3: Architecture of the proposed model. Adapted from [3].

Parameter	Description	Setting
C	The number of hyper-spectral bands in the hyper-spectral image.	51
S	The size of the neighborhood window.	11
L	The size of the latent dimension	40

Table 1: Hyperparameters to the Model

3.1 Local Sensing Network

Local sensing is preformed using stacked CNN. The goal of this CNN is to extract spatial information from all the input bands in neighborhood region around the input pixel. Three convolution layers are used for our model. The size of the kernels used and the number of filters for each CNN layer are shown in Table 2. Each layer is created using a PyTorch Conv2d layer with default values used for all non-required parameters. The final output of this network is a vector with $L/4$ encoded spatial features, (x_{ls}).

Layer	Kernel Size	Result Size	Number of Filters
1	$\lfloor S/2 \rfloor + 1$	S_1	L
2	$\lfloor S_1/2 \rfloor + 1$	S_2	L
3	S_2	1	$L/4$

Table 2: CNN Layers

3.2 Sequential Sensing Network

Sequential sensing attempts to encode features that are sequentially related in the spatial domain. The motivation behind this is that the spatial information surrounding a pixel is most often interpreted from left to right, and top to bottom. An LSTM network can capture relationships that occur over a sequence of input data using memory cells. The input for this network is an SxS matrix of pixel vectors in the neighborhood region, which are flattened into a sequence of S^2 pixel vectors that get passed into a stacked LSTM network with three layers. The middle layer has a hidden dimension equal to the latent dimension, and the final layer shrinks the hidden dimension down to the expected output size of $L/4$. Since an LSTM operates on sequences, an average pooling layer averages the output of the network across the sequence to obtain the final sequential feature vector. The result of this network is an encoding of the sequential spatial features in the neighborhood around the target pixel vector, denoted (x_{ss}).

3.3 Spectral Sensing Network

Spectral sensing encodes features throughout each of the hyper-spectral bands. There are relationships between the spectral bands that can be seen when looking at each band individually. When looking at a hyper-spectral image containing text, the text is visible to varying degrees throughout each spectral band. It may be the case that in some spectral bands, the textual information exists just beyond the capability of human perception. Alternatively, there may be small remnants of textual information that can be pieced together from multiple spectral bands. Therefore it is important to capture this potential information in any latent representation of a pixel.

The spectral sensing network is an encoder network that outputs the mean (μ_{partial}) and variance (σ) vectors that parameterize a latent distribution for an input pixel vector. The input to this model is a single pixel vector comprising C hyper-spectral channels. This is passed through a feed-forward network with three layers (input, hidden, and output). The input layer is the same dimension as the pixel vector, the hidden layer is the size of the latent dimension L, and the final output dimension is equal to L for the variance vector and $L/2$ for the mean vector. All layers in the encoder are activated with a ReLU activation function.

3.4 Spectral-Spatial Encoder

Spectral and spatial features are then combined together to form a spectral-spatial encoder. The spatial features from the CNN and LSTM networks are concatenated[1] with the mean vector from the spectral sensing network to form a revised latent mean vector that has the same size as the latent dimension, L. The final results of this encoder are a mean and variance vector that can be used to parameterize a normal distribution over the latent feature space for the input pixel vector. This distribution can be sampled to obtain a specific latent vector that can be used for an analysis task or fed into a decoder network that attempts to recreate the original image.

$$\mu = x_{ls} + x_{ss} + \mu_{\text{partial}} \tag{4}$$

3.5 Spectral-Spatial Decoder

The decoder is an important part of the VAE framework. This network takes a latent representation for the input and attempts to recreate the original input. This step is important because a better latent representation will allow the decoder to better reconstruct the input, and the reconstructed input provides a way to compute an appropriate loss function. The decoder network consists of three layers. The first two are the size of the latent dimension, and the output layer is the size of the original input. The first two layers use a ReLU activation function, while the output layer uses a sigmoid activation. Notice that this network is a mirror of the encoder network, though it is allowed to learn different weights than the encoder network.

[1] See Equation 4

3.6 Loss Function

Neural networks rely on a loss function to learn what the optimal values of their weights should be. A good loss function fully captures all relevant details of the performance of the network, and then backpropagation can be used to update the weights for the network components. Yu et. al. propose a loss function for this model with three parts, each to capture a different aspect of the model's performance.

The first part is reconstruction loss term shown in Equation 5. This measures the accuracy of the decoder's reconstruction of the original input pixel using the latent representation.

$$\Gamma_{\text{R}} = \sum_{i}^{L} (x_i - \hat{x}_i)^2 \tag{5}$$

Both the CNN and LSTM networks are designed to capture spatial information in the neighborhood region around the target pixel vector. Therefore, it is reasonable to expect that the information encoded is similar. This notion of similarity into the loss function with a KL divergence term between the two spatial vectors shown in Equation 6.

$$\Gamma_{\text{homology}} = \frac{1}{2} \sum_{i}^{L/4} \left(x_{ls_i} \log \frac{x_{ls_i}}{x_{ss_i}} + x_{ss_i} \log \frac{x_{ss_i}}{x_{ls_i}} \right) \tag{6}$$

The final term in the proposed loss function is a KL-divergence term designed to bring the distribution parameterized by μ and σ closer to a normal distribution $\mathcal{N}(0, 1)$. This term is shown in Equation 7.

$$\Gamma_{\text{KL}} = \sum_{i}^{L} (\sigma^2 + \mu^2 - \log(\sigma^2)) \tag{7}$$

The final loss value is the combination of the three loss terms as shown in Equation 8.

$$\Gamma = \Gamma_{\text{R}} + \Gamma_{\text{homology}} + \Gamma_{\text{KL}} \tag{8}$$

4 Methodology

4.1 Data

To validate the model we created works as intended by the authors of [3], we have used the same datasets as described. Yu et. al. use hyper-spectral aerial imagery from three locations: Indian Pines, Pavia University and Salinas Valley and we add an additional validation dataset, Pavia. Each dataset was collected at different times, with different sensors, resulting in differing image sizes and different bands. The authors are looking to classify different regions and objects depending on the dataset. For example the dataset from Indian Pines contains imagery of corn, grass, woods and more. The authors of [3] performed classification on these different regions/objects using a SVM.

We have worked on a multi-spectral image of a reagent treated document with 52[2] bands. This specific document is Manuscript 38 from the Biblioteca Capitolare in Verona. Not only has document 38 had reagent applied to it, its also a palimpsest. Text has been removed from document 38 at least 4 times. We have multi-spectral image data for four other documents that can be used for future work. Additionally, we have partially recovered documents with letters that were manually traced from what was already visible in the original text. The combination of this data can be used to train and evaluate our VAE on the task of recovering reagent obscured text.

4.2 Experiments

We perform experiments to evaluate our encoding method by comparing to baselines of the manually recovered letters as well as using the method proposed by LaLena on the spectral information. If

[2]Only 51 bands are usable as one band was corrupted.

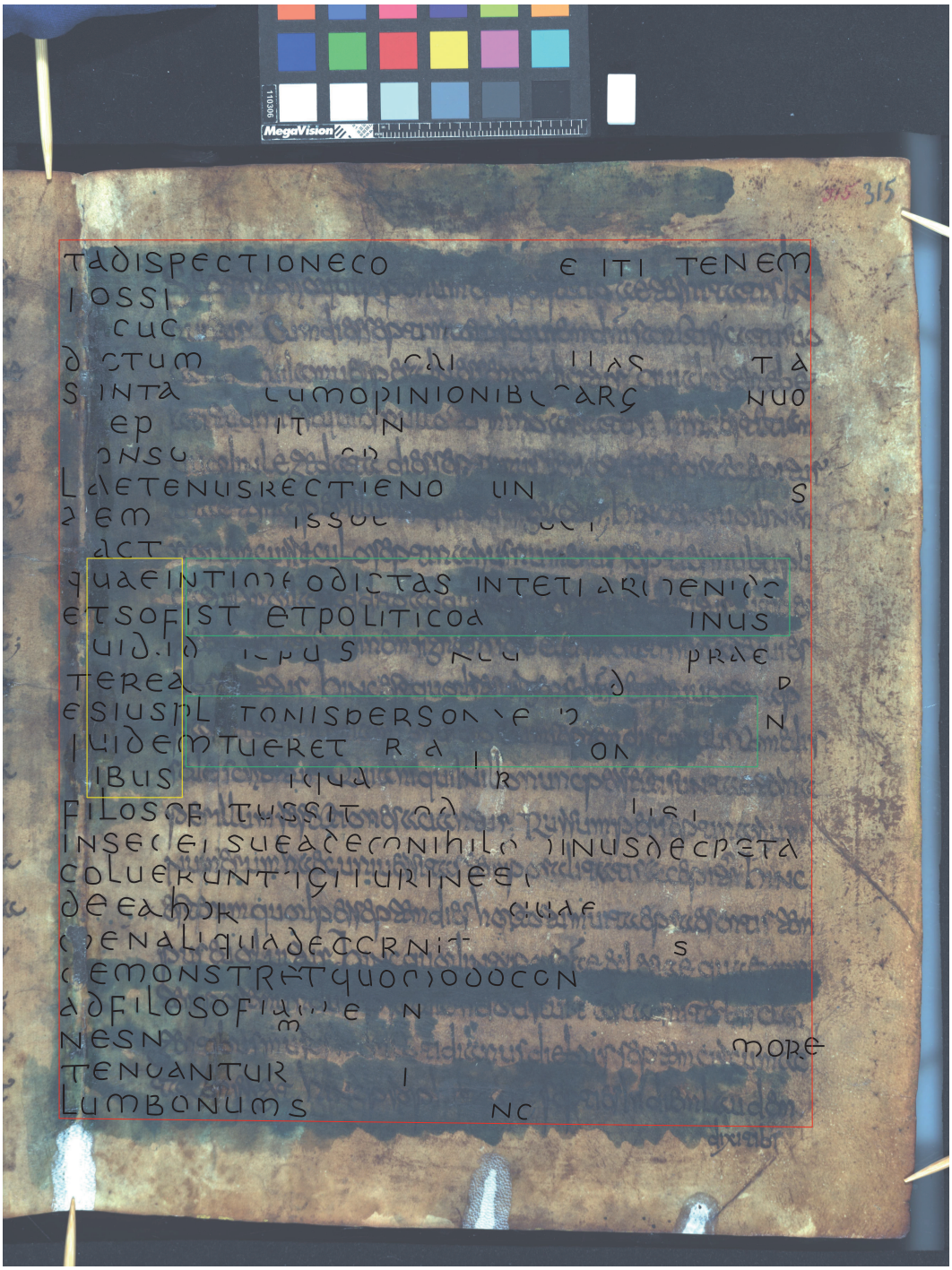

Figure 4: Document containing manually recovered letters.

our method succeeds in recovering more letters as well as the same letters as the manually recovered text or LaLena, we can consider U_{Hfe}SRVAE as a plausible method in recovering text from reagent treated documents. See Figure 4 for an example of manually recovered letters on a document.

To ensure the model was preforming properly we first reproduced the classification results from [3] using the latent dimensions encoded by the VAE to train a classifier using KNN. Using the same datasets as in [3] we were able to replicate the model, and the directly compare the authors' results

with our own. The only notable change to Yu et. al.'s methodology for this experiment is our use of the K-nearest neighbors algorithm instead of a tuned SVM for classification of the pixels.

The decoder should be able to create images that look similar to the original. In order to ensure the decoder, and therefore the encoder, was functioning properly we also compare decoded bands with the original bands to qualitatively evaluate reconstruction.

We also use an attempted to use a simple threshold method proposed by LaLena. Encoded images were added together resulting in an image representing a combination of the encoded results. Any pixel above a given threshold in this combination image was classified as text.

The last experiment performed is a K-means clustering on the latent representation of the image. Our approach is to generate latent representations for each pixel in the target hyper-spectral image using U_{Hfe}SRVAE . This creates a new image that is encoded in the latent dimension. We then use K-means clustering to group every pixel into one of two clusters. The thought is that one cluster will emerge as textual information, and the other cluster will be the background noise of the document. Using the cluster label for each pixel, the image is colored to showcase the clustering result.

4.3 Results

Figure 5 shows a comparison of the results from the Indian Pines dataset from [3] and the results from our replication of the model.

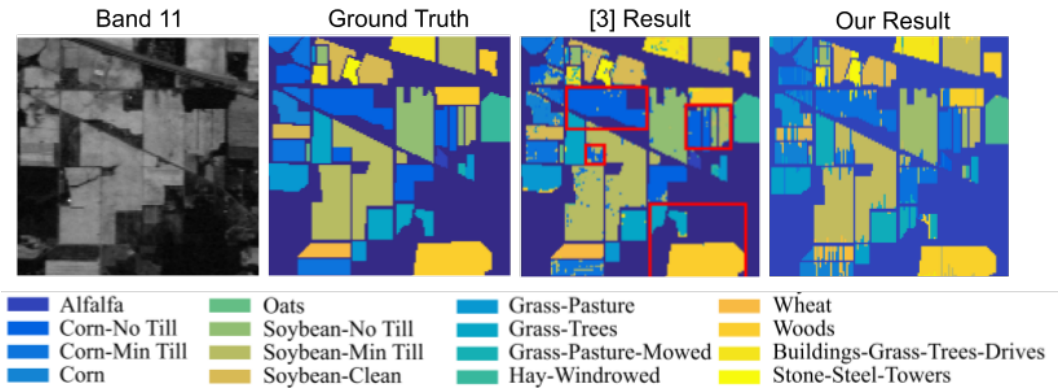

Figure 5: Comparison of classification of regions/objects in an image from the Indian Pines dataset. An image of band 11, the ground truth, results from [3] and our results are shown. Note that our results do not match the color key from [3], though the colors are similar and the reader should be able to match the original color key with the colors in our result.

The reconstructed results, as expected, are not perfect due to the Gaussian noise. The reconstructed images look like nosier versions of their original counterparts. This can be seen in figures 6 and 7.

The simple threshold approach to detecting text does not give desired results. As seen in figure 8, the detected regions are very sparse. Decreasing the threshold causes over detection and most of the image is classified as text.

The K-means method of detecting text results in much better detection of characters, as seen in figure 9. There are still false positives and negatives, but overall the detection is improved by using K-means over a simple threshold. Though our proposed method does not work as well as LaLena's method on sections of document heavily treated with reagent, as seen in figure 10.

4.4 Analysis

Our implementation of the model proposed by [3] Yu et. al. appears to be sufficient. From a qualitative view, the results our model achieves are visually similar to the results achieved in the paper. On the Indian Pines dataset, our model achieved an accuracy of 80%. This result falls slightly short of the overall accuracy that Yu et. al. obtained (91%), but it does indicate that the model is

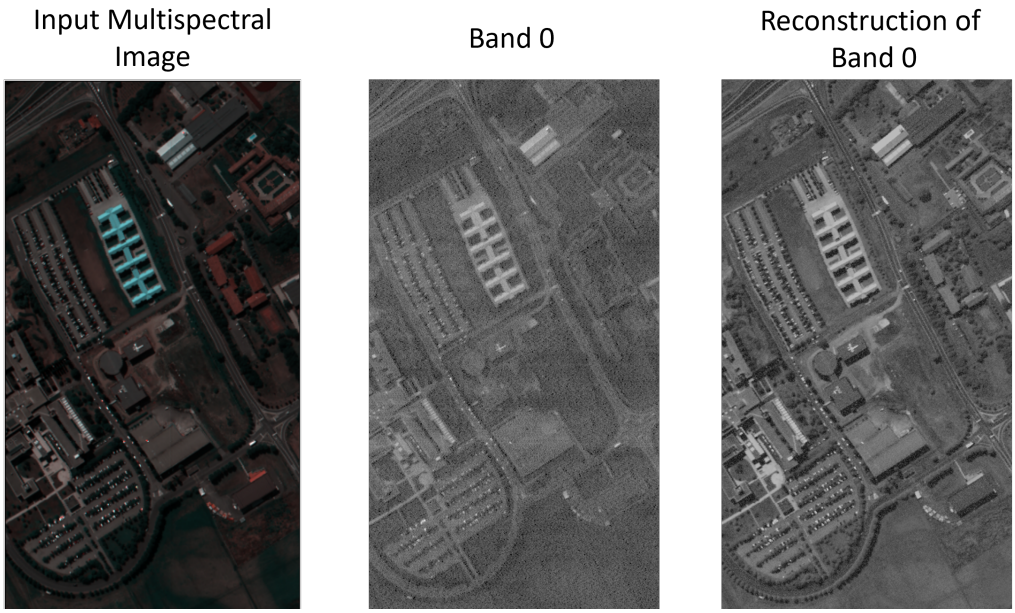

Figure 6: A comparison of the original first band in the multi-spectral data with the reconstructed result of that same band from the decoder. The input multi-spectral image is a pseudo color representation of the multi-spectral data. This image is from the Pavia University data set.

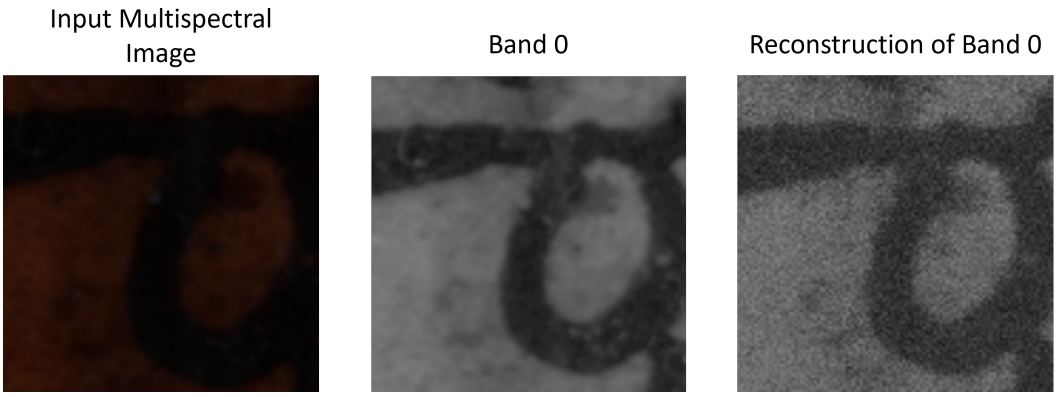

Figure 7: A comparison of the original first band, representing 615nm, in the multi-spectral data with the reconstructed result of that same band from the decoder. The input multi-spectral image is a pseudo color representation of the multi-spectral data. This image is a small subsection of document 38.

working similarly to the intended solution and could be due to our use of a different classification head.

We are also able to obtain visually similar reconstructions for each hyper-spectral layer, albeit with some noise. Various reconstruction samples can be seen in figures 6 and 7.

Lastly, it appears that the latent space learned by the U_{Hfe}SRVAE model is not compatible with the text thresholding approach used by LaLena as shown in figure 8. This is likely due to the noisy nature of the latent images. The result obtained in Figure 2 shows a much tighter pseudo-coloring using threshold than the results obtained with the latent representation produced by U_{Hfe}SRVAE .

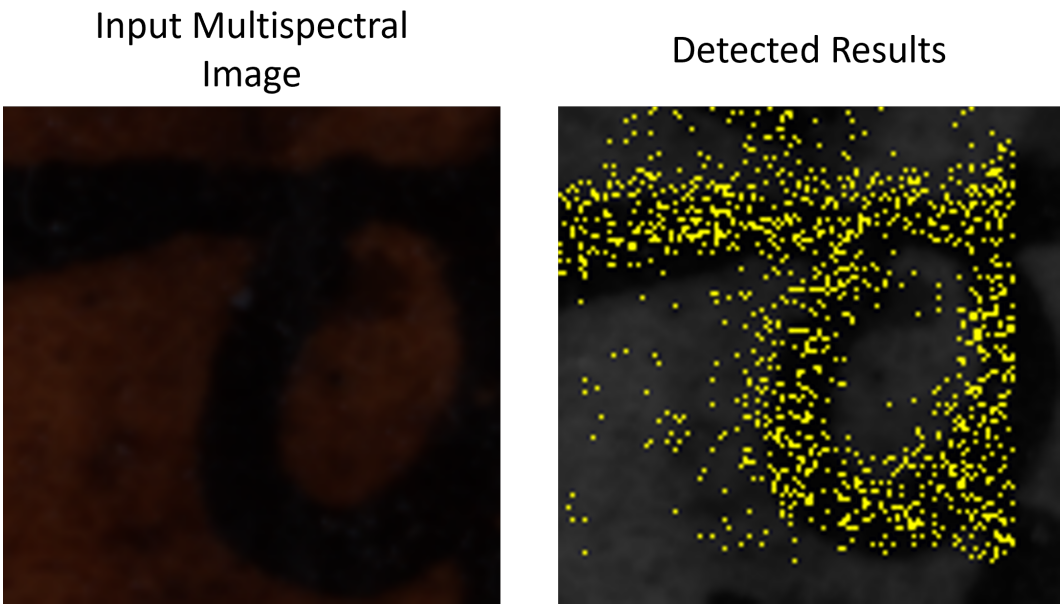

Figure 8: A comparison of a pseudo color representation of the multi-spectral data of document 38 and the detection results from a simple thresholding method. The threshold was set to 3.

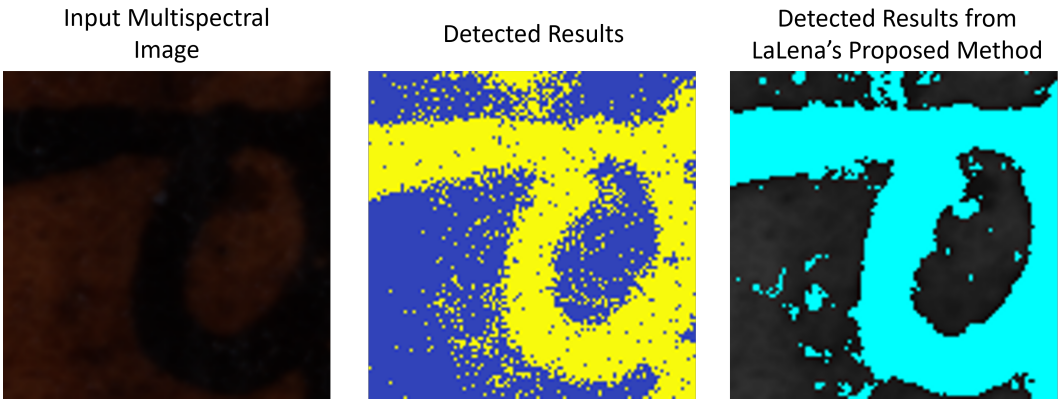

Figure 9: A comparison of a pseudo color representation of the multi-spectral data of a subsection document 38 and the detection results the K-means clustering method. The result from LaLena's proposed method are also added for comparison. This subsection is treated with little to no reagent.

5 Conclusion

Our experiments indicate that U_{Hfe}SRVAE does not provide a better latent representation of palimpsest documents with respect to the approach used by LaLena. The latent representation appears to produce random noise when combined with the thresholding method of LaLena. However, we were able to reproduce the model proposed by [3] with an accuracy of 80% on the Indian Pines dataset used by the original paper, suggesting that the poor results on document text recovery are not the result of a bug in the model.

The latent images from the proposed method does not work well with a simple thresholding technique. This is because the latent images look too "noisy" for thresholding to work. Only some pixels in a region will overcome the threshold and be correctly classified do to the noisy nature of the latent images.

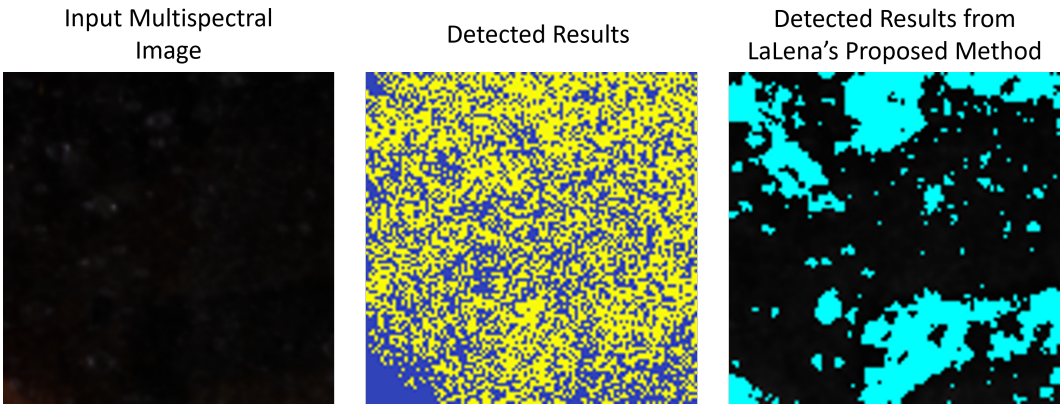

Figure 10: A comparison of a pseudo color representation of the multi-spectral data of a subsection document 38 and the detection results the K-means clustering method. The result from LaLena's proposed method are also added for comparison. This subsection is heavily treated with reagent.

Dataset	Classification Accuracy
Salinas	97.23
PaviaU	86.53
Pavia	92.96
Indian Pines	87.96

Table 3: Classification accuracy for each dataset in the original paper [3]. Accuracy is computed relative to the ground truth, and classification is performed using KNN.

Additionally, U_{Hfe}SRVAE provides a sufficient latent representation that has good visual results for text identification when using K-means clustering. Our results show that K-means clustering can cluster text in the latent representation of the hyper-spectral image when it is visible to the human eye. On a more difficult problem, where the text is not visible to the human eye in the original hyper-spectral image, the K-means clustering approach still struggles to detect potential textual artifacts, as seen in Figure 10, Figure 11, and Figure 12.

An interesting avenue for future work can be found in the reconstructed images. Since the latent representation that is used for reconstruction is sampled from a distribution, there is a possibility that there exists a reconstructed sample that contains legible text in one of the hyper-spectral bands. Future experiments would attempt to sample a wide range of the distribution and see if these results

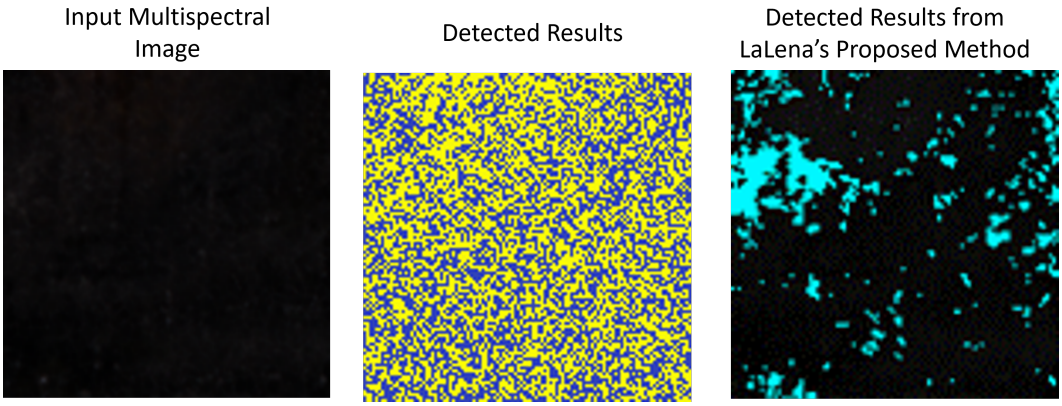

Figure 11: Comparison of textual clustering methods for a subsection of document 38 where the text is clear to a human observer. The k-means method and LaLena's method perform similarly.

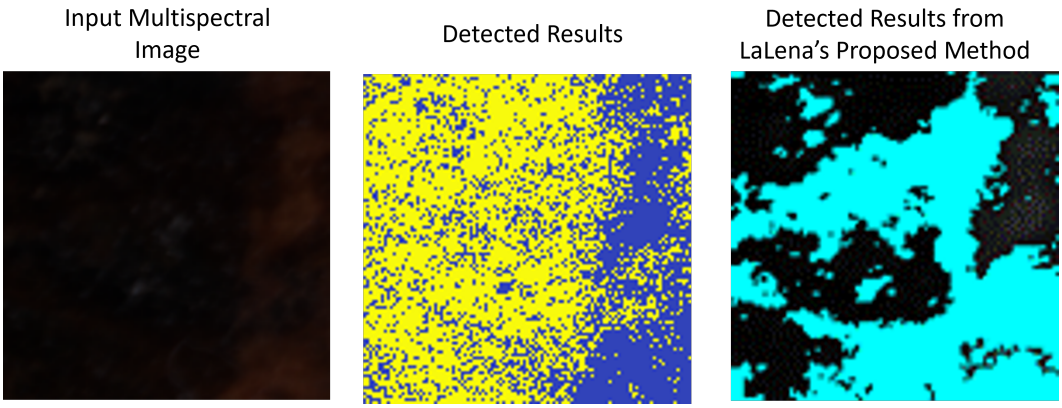

Figure 12: Textual clustering results for a subsection of document 38 where the original text is not easily visible.

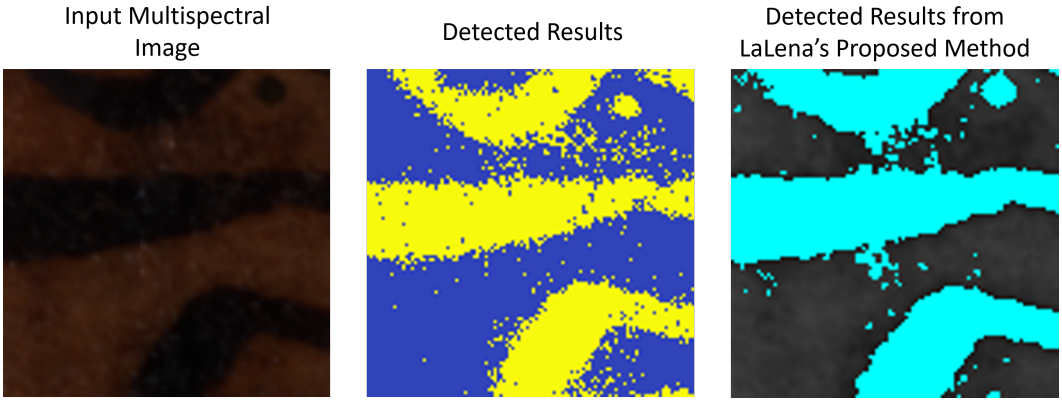

Figure 13: Textual clustering results for a subsection of document 38 where the original text is easily visible.

appear; if they do, then research would also be needed to determine the likelihood that the text is accurate to the original document, and not a meaningless artifact of noise.

References

[1] Albrecht, F. (2012). 'Between Boon and Bane: The Use of Chemical Reagents in Palimpsest Research in the Nineteenth Century'. Care and Conservation of Manuscripts, 13, 147.

[2] Bei Grace Cao, & David W. Messinger (2022). Graph convolutional network for automatic pigment clustering of cultural heritage artifacts. In Algorithms, Technologies, and Applications for multi-spectral and hyper-spectral Imaging XXVIII (pp. 120940U). SPIE.

[3] Yu, W., Zhang, M., & Shen, Y. (2021). Spatial Revising Variational Autoencoder-Based Feature Extraction Method for hyper-spectral Images. IEEE Transactions on Geoscience and Remote Sensing, 59(2), 1410-1423.

[4] Leidy P. Dorado-Munoz, & David W. Messinger (2015). Schrodinger Eigenmaps for spectral target detection. In Algorithms and Technologies for multi-spectral, hyper-spectral, and Ultraspectral Imagery XXI (pp. 947211). SPIE.