

# Interactive Visualization of Hyperspectral Images based on Neural Networks

Feiyu Zhu<sup>1</sup>, Yu Pan<sup>1</sup>, Tian Gao<sup>1</sup>, Harkamal Walia<sup>2</sup>, Hongfeng Yu<sup>1</sup>

<sup>1</sup> Department of Computer Science and Engineering, University of Nebraska-Lincoln

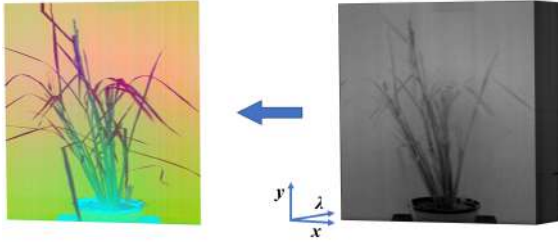
<sup>2</sup> Department of Agronomy and Horticulture, University of Nebraska-Lincoln

**Abstract**—It is challenging to interpret hyperspectral images in an intuitive and meaningful way, as they usually contain hundreds of dimensions. We develop a visualization tool for hyperspectral images based on neural networks, which allows a user to specify the regions of interest, select bands of interest, and obtain hyperspectral classification results in a scatterplot generated from hyperspectral features. A cascade neural network is trained to generate an scatterplot that matches the cluster centers labeled by the user. The inferred scatterplot not only shows the clusters of points, but also reveals relationships of substances. The trained neural network can be reused for time-varying hyperspectral data analysis without re-training. Our visualization solution can keep domain experts in the analytical loop and provide an intuitive analysis of hyperspectral images while identifying different substances, which are difficult to be realized using existing hyperspectral image analysis techniques.

■ **HYPERSPECTRAL CAMERA** takes images of objects at different wavelengths [1] and can provide abundant spectral information about different objects, thereby being widely applied in many disciplines, such as remote sensing and plant science. Hyperspectral images can be modeled as a hyperspectral cube (Figure 1). However, they are different from traditional volume data, where the spatial dimensions and the spectral dimension have different physical meanings. The  $x$  and  $y$  dimensions are the spatial axes of the objects in the images. The  $z$  or  $\lambda$  axis is the spectral axis, which contains spectral information of substances. Here we define substances as objects that have the

same spectral characteristics. Each 2D position in the  $xy$ -plane corresponds to a hyperspectral curve that is formed by a series of pixels from the same 2D position of all images along the  $z$  or  $\lambda$  axis.

Many existing studies investigate how to identify substances from hyperspectral images through clustering, classification, or image fusion techniques [2], [9], but could not meet emerging requirements on finer-grained analysis. For example, in our collaboration with plant scientists, we have collected time-varying hyperspectral images of plants. Our collaborators, who are domain experts in the field of agronomy and horticulture, want to study how to differentiate various parts of



**Figure 1.** Illustration of hyperspectral images. Hyperspectral images (right) of a rice plant is taken over a series of bands and a fused image (left) is usually generated as the final result.

an object with different bands, leading to new analysis requirements:

- Extract essential hyperspectral features that can well represent the hyperspectral images.
- Support interactive exploration of substances with classification or image fusion results.

However, most traditional methods [2], [4], [9] act as black boxes and generate one-time results from all the bands of a whole image, which are less intuitive and flexible for further investigating hyperspectral features. Meanwhile, these methods often lack a support of interactive exploration.

Through detailed discussions with our domain experts, several challenges have been identified to meet these requirements. First, domain experts want to interactively select regions of interest (ROIs) and bands of interest (BOIs). While it is relatively easy for users to define ROIs on a 2D image, it is non-trivial to develop an intuitive visualization interface to select BOIs from hundreds or thousands of bands with significant intensity variations. Second, the colors of image fusion results often change according to different user selections of regions or bands, making it challenging for users to identify and track the same substances, particularly with time-varying data. Third, it is difficult for users to characterize features from a large number of hyperspectral bands and identify the correspondence between hyperspectral features and image fusion results.

To address these challenges, we advocate combining classification and visualization to develop an interactive analytics pipeline for hyperspectral images. Our work makes the following major contributions:

- We explore and identify appropriate hyperspectral information to support interactive selections of ROIs and BOIs.
- We devise a new neural network based approach to generate a scatterplot and facilitate users to interactively examine the correspondence between hyperspectral features and substances by brushing the scatterplot.
- We use the neural network to generate pseudo colors and lead to stable image fusion results for identifying and tracking substances.

We have demonstrated the effectiveness of our approach using datasets from remote sensing and plant phenotyping. Our visualization solution keeps domain experts in the analytical loop. It can facilitate scientists from different domains to effectively study hyperspectral images and gain new discoveries that are not conveyed with existing techniques.

## Related Work

Hyperspectral images have been studied using different approaches. We categorize the existing hyperspectral image analysis methods into three groups, clustering methods, classification methods, and visualization methods, and summarize the existing work as follows.

### Clustering Methods

As substances are characterized by different hyperspectral curves, an intuitive solution is to cluster hyperspectral curves according to their hyperspectral shapes. The most commonly used clustering methods include  $k$ -means clustering, hierarchical clustering, and mean-shift clustering. Euclidean distance, Procrustes analysis, Spectral Angular Mapping have been used as distance measures for these clustering methods. However, hyperspectral images can be affected by uneven illumination [6], where hyperspectral curves that belong to the same substance can have the same hyperspectral shape but different scales. Thus, clustering methods often incorrectly classify the hyperspectral curves with small scales. In this work, we advocate a combination of neural networks and visualization methods instead of clustering methods to identify substances with uneven illumination.

## Classification Methods

Classification determines which substance a pixel belongs to and usually requires training labels. Machine learning has been widely used for classification of pixels in hyperspectral images. Maximum likelihood classification, support vector machine (SVM), and convolutional neural network (CNN) are commonly used methods for classification [2]. Neural networks can handle the problem of uneven illumination as they can be trained to recognize scaled hyperspectral curves of the same substance. However, classification methods only give users inferred labels and most act as black boxes. It is difficult for users to observe the distribution of the hyperspectral features or examine how substances are similar or different in terms of hyperspectral characteristics. To address these issues, we develop new visualization capacities to enable an interactive exploration of hyperspectral features and substances.

## Visualization Methods

Visualization is used to find a representation that keeps or enhances hyperspectral information in images. Image fusion is a widely used method for hyperspectral images, where the original multi-dimensional hyperspectral images are fused into one color image after dimension reduction and mapping of pseudo colors [9]. Some useful dimension reduction methods [4] include Principal Component Analysis (PCA), Independent Component Analysis (ICA), Linear Discriminant Analysis (LDA), and t-Distributed Stochastic Neighbor Embedding (t-SNE). However, if the differences between two substances are relatively small compared with the global maximum differences, these substances will be very close in the projection. Hence image fusion can also suffer from uneven illumination. Image fusion with pseudo-colors can only be treated as a basic type of visualization. Many other visualization techniques remain to be applied on hyperspectral images.

Cui et al. [3] developed a visualization tool for image fusion using convex optimization. Kim et al. [8] developed an interactive visualization tool for hyperspectral images of historical documents based on image fusion. Some work attempted to generate fused images ready to be visualized on a display [10], but did not directly deal with the visualization of hyperspectral images.

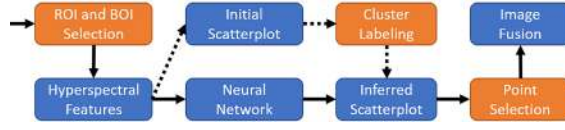
One key issue in the existing work is that the color of the same substance can be changed with different user selections in hyperspectral images. We want the visualization results to be stable where the same substance always has the same color. This is especially desired for the study of time-varying hyperspectral images. We also want to create a correspondence between a feature space and an image space, which is not conveyed in the current visualization methods. In our design, instead of inferring labels, we use a neural network to infer the positions of hyperspectral features in a scatterplot. The neural network is no longer used for classification but visualization. The neural network can create stable visualization results once it is trained. The inferred scatterplot helps users observe the distribution of hyperspectral features and understand the similarities among substances. The selection of points in the scatterplot is used to generate corresponding image fusion results where the pseudo colors are generated by the neural network and are resistant to uneven illumination.

## Rationale

We develop an interactive visualization tool to help scientists study hyperspectral images. We summarize how we can make use of the global statistics and visualization of dimension reduction to design an interface. We first allow a user to interactively select ROIs and BOIs. Then, we calculate the hyperspectral features and use these features to generate an initial scatterplot. Numbered circles are shown at the center of each cluster of labeled data in the scatterplot. As the initial scatterplot may not be optimal, users will define a target scatterplot by leveraging their domain knowledge to drag numbered circles to new positions interactively. After user definition, a neural network will be trained to generate an inferred scatterplot that matches the user-defined scatterplot. Finally, the user can interactively and intuitively select substances in the inferred scatterplot and obtain corresponding image fusion results for substances in the hyperspectral images. [Figure 2](#) shows the pipeline of our interactive tool.

## Region and Band Selection

In the traditional methods, all the bands and the whole image are usually used for image fusion. However, analysis of the whole image may shadow



**Figure 2.** The pipeline of our tool. Orange blocks indicate user interaction. The dashed arrows mean the steps that are followed once.

local details and the selection of all the bands may be redundant. Thus, it is necessary for users to interactively select ROIs and BOIs.

ROIs can be easily defined by users selecting regions on a 2D image. However, the selection of BOIs is less intuitive. In order to guide a user in the selection, we want to extract certain global information of hyperspectral images by deriving metrics to evaluate each image. After consulting with our collaborators who are the domain experts in agronomy and horticulture, we noticed that the bands that they are usually interested in correspond to images with high contrast. Such an image usually contains more information than images with low contrast. Thus, we choose image entropy and intensity mean as two metrics for identifying images with appropriate contrast. Entropy  $H$  of an image can be defined as  $H = -\sum_{i=0}^{255} p_i \log(p_i)$ , where  $p_i$  is the probability of an intensity value  $i \in [0, 255]$  in an image. As the user usually tends to select the peaks in a curve [12], the entropy curve is effective in helping users select BOIs.

Other metrics can also be used to evaluate an image. For example, the variance can be used to evaluate the distribution of pixel values in an image. Cross-correlation and mutual information can be used to evaluate the change of consecutive images. In our design, we choose entropy empirically to guide users in the BOI selection.

### Feature Derivation

To help users observe the correspondence between hyperspectral features and image fusion results, we map the hyperspectral curve of each pixel to a point in an inferred scatterplot using neural networks. Although neural networks can extract important features from original hyperspectral data, we found that better results can be obtained with extra pre-defined statistical features in our experiment.

To evaluate hyperspectral curves, many sta-

tistical metrics can be used. These metrics can also be applied for the original data or derivatives of hyperspectral curves. The empirically selected metrics and derivatives are listed as follows:

- Metrics: mean, variance, skewness, and kurtosis.
- Derivatives: first and second derivatives.

These derived features can be used with the original data to characterize the spectral curves in a non-linear way [5]. The calculation of hyperspectral features can be done in parallel due to independence of pixels. Thus, GPUs can be used to accelerate the computation. We allow the user to choose the hyperspectral features that will be used as the input of a neural network. Although we provide many features and some of them may be redundant, the neural network can find important features by assigning different weights to them during training, which will be discussed in the Network Training section.

### Scatterplot Definition

We want the points in a scatterplot to form distinct clusters so that the user can easily select the clusters to study different substances. However, such a scatterplot may not be readily available until the neural network is trained. We want the user to specify the locations of the clusters and use the locations as the training target.

We provide an initial 2D scatterplot using PCA. For each group of labeled data, we draw a numbered circle indicating the cluster center in the scatterplot. Users can then drag the circles interactively based on their domain knowledge to sparsely distribute these circles in the scatterplot. The positions of the user-defined circles will be used as the target for training a neural network that will infer a new scatterplot.

As the input to the neural network is the features of each pixel, we empirically chose to use a fully connected cascade neural network [7]. Skip connections are used in the cascade neural network. The same kind of idea has been widely adopted in many convolutional neural networks to improve the results.

### Network Training

The network has two hidden layers, and the numbers of units are empirically set as 10 and 3





**Figure 3.** The design of our interface. Each image corresponds to one step in the pipeline. Dashed arrows mean that the steps are followed only once.

for the first and second hidden layers, respectively. The input of the network will be the derived hyperspectral features at each 2D position, and the output of the network will be a 2D position in the inferred scatterplot. The training target is the corresponding cluster center, which is defined by the user after re-positioning the circles in the initial scatterplot. The maximum number of epochs for training is set to 2000. The actual number of epochs that the network is trained for may be less, as the training will stop early if the result does not significantly improve anymore. To reduce overfitting, the input data are split so that 70% is used for training, 15% for validation, and 15% for testing. The features extracted in the last hidden layer of the neural network will be used as the pseudo-colors of the inferred scatterplot and the final visualized image.

## Interface Design

We have designed our interface using MATLAB that supports user interactions in multiple steps. Figure 3 illustrates the interface and windows of the steps, where a public dataset Kennedy Space Center (KSC) is used as an example.

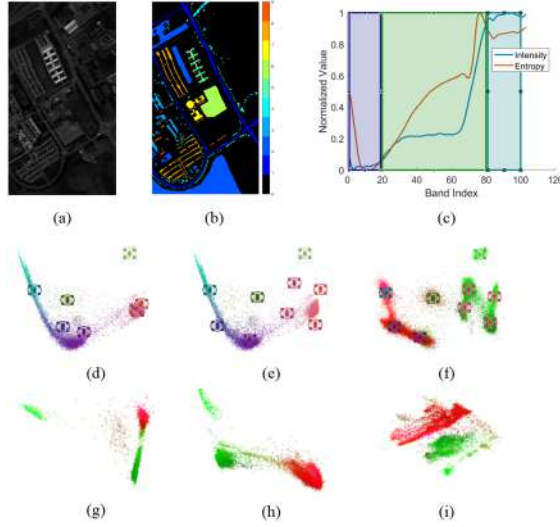
A user can select one of the hyperspectral images to be shown in a window, where the user can draw rectangular boxes to select ROIs. After the selection of ROIs, the corresponding entropy curve is shown in a plot, where the user can select BOIs. Note that a selected range of the entropy curve corresponds to a selection of the original hyperspectral images. After the selection

of ROIs and BOIs, a set of hyperspectral features selected by the user will be calculated for BOIs. An initial scatterplot is generated using PCA. In the initial scatterplot, circles are drawn for individual groups of labeled data, where the center of a circle is the mean position of the group of points. In an ideal scatterplot, different clusters should be well distinguished, while the points in each cluster should be as close to the cluster center as possible. However, the initial scatterplot may not be desired as some circles overlap others, as illustrated in Figure 3. Users can leverage their domain knowledge to drag the circles to new positions. The new positions of the clusters are used as the training target for the neural network. When the cascade neural network is trained, a scatterplot can be inferred, which has a better separation of different clusters than the initial scatterplot. An example is shown in Figure 3.

The user can freely select points in the scatterplot by drawing polygons, and the corresponding hyperspectral pixels are visualized as shown in Figure 3. For image visualization, we also use the features of the last hidden layer as the pseudo-colors. Based on the correspondence between the scatterplot and the visualized image, the user can interactively and intuitively explore substances and study their details. To help the user obtain consistent results, we allow the user to define ROIs, BOIs, and polygons using parameters and load pre-defined scatterplots and pre-trained networks.

**Table 1. The sizes of different datasets. Note that not all the pixels are labeled.**

Dataset	Image Sizes	#Labeled Pixels
PaviaU	610*340*103	42776
Sorghum	420*320*244	144551
Maize	420*320*243	107319



**Figure 4.** The result of Pavia University.

## Results

In this section, we show how our interface can be used to explore hyperspectral images interactively. We have tested our tool on a set of datasets from several domains, including the public datasets in remote sensing and the datasets collected by our domain experts in plant science. The image sizes and the number of labeled pixels of three datasets are listed in Table 1. The results of other datasets are not shown due to the page limit.

### Remote Sensing

Hyperspectral images have been widely studied in remote sensing. These hyperspectral images are usually taken by satellites or high-flying aircraft, and regions in the groundtruths are manually labeled. The labels may not be completely correct in terms of hyperspectral characteristics as one region may not be homogeneous and can contain many substances. Thus, it is challenging to cluster the pixels of each labeled region correctly.

Figure 4<sup>1</sup> shows the result of a public dataset

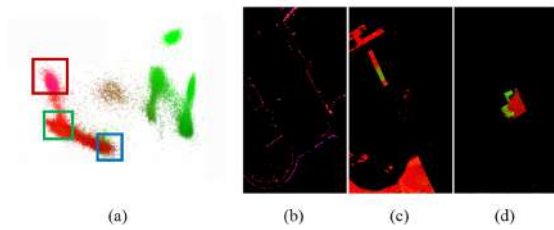
<sup>1</sup>Figures 4, 6 and 7 use the same layout. Without specification, the scatterplots in our examples only contain the points corresponding to labeled pixels.

Pavia University with nine labels: Asphalt, Meadows, Gravel, Trees, Painted metal sheets, Bare soil, Bitumen, Self-blocking bricks, and Shadows. Figure 4(a) is one of the original hyperspectral images. Figure 4(b) is the groundtruth with pseudocolors. The colorbar in the image shows the correspondence between the labels and pseudocolors. In this and following examples, the whole images are selected as the ROIs. Figure 4(c) shows the selection of three BOIs using our interface.

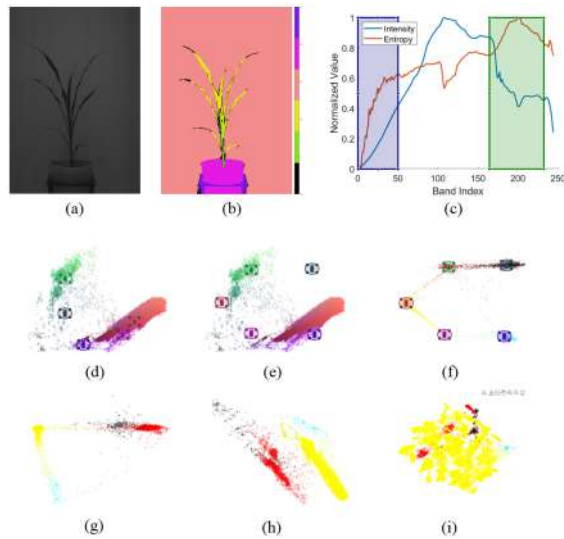
Figure 4(d)(e)(f) are the initial scatterplot, the user-defined scatterplot, and the inferred scatterplot, respectively. The colors in Figure 4(f) are obtained using the features of the last hidden layer of the cascade neural network to show the correspondence between points in different scatterplots. Our design can allow a user to select different substances to gain intuitive visualization results. It can be seen that the inferred scatterplot shows distinct clusters of points that match the numbered circles defined by the users in the user-defined scatterplot. The points scattered between clusters are usually boundaries between substances in the original hyperspectral images.

The scatterplot generated in Figure 4(f) clearly shows these nine substances based on domain knowledge. Clusters four, two, and six are labeled as Trees, Meadows, and Bare soil, respectively. In our common sense, the region of Trees contains more leaves while the region of Meadow contains partially leaves and partially soil. Thus, Trees should be close to Meadows, and Meadows should be close to Bare soil. This knowledge is revealed in Figure 4(f), where cluster four connects cluster two and cluster two connects cluster six. Similarly, Gravel, Self-block bricks, Asphalt, and Bitumen are all products made from rocks and should have similar characteristics. Thus, clusters three, eight, one, and seven are relatively close to each other in Figure 4(f). We define the relationship as the relative positions of different substances. To the best of our knowledge, this type of relationship between substances in hyperspectral images has never been presented in the previous work.

Figure 4(g)(h)(i) are the results generated using ICA, LDA, and t-SNE, respectively. Similar to Figure 4(f), their colors are obtained using the features of the last hidden layer of the cascade neural network. We can see that Figure 4(g)(h)(i) cannot clearly reveal clusters of points. The



**Figure 5.** Selections of points in the inferred scatterplot of dataset Pavia University. (a) The inferred scatterplot where the user has made three different selections. (b)(c)(d) are the image fusion results corresponding to the points in the red, green, and blue rectangles in (a).



**Figure 6.** The result of a sorghum plant.

existing machine learning based methods [2] can generate labels similar to Figure 4(b). However, they cannot tell the user if two labels are similar or not in terms of hyperspectral characteristics.

With our tool, the user can further generate a fused image interactively by selecting points in the inferred scatterplot. For example, three different clusters are selected in Figure 5(a). Figure 5(b)(c)(d) show the visualization of Trees, Meadows, and Bare soil, respectively. The shape of each classified region is close to the groundtruth.

## Plant Science

Hyperspectral imaging techniques for plants have been available in recent years. Figure 6 shows the result of a sorghum plant. Here, two BOIs are selected empirically, as shown in Fig-

**Table 2.** The MSE ( $10^{-3}$ ) of the inference using different inputs.

Feature Type	Org.	1st	2nd	Org.+1st+2nd
HyperCurve	4.909	7.795	10.576	0.526
HyperMetrics	5.102	13.268	29.418	0.440
HyperCurve+HyperMetrics	4.421	7.306	10.14	<b>0.426</b>

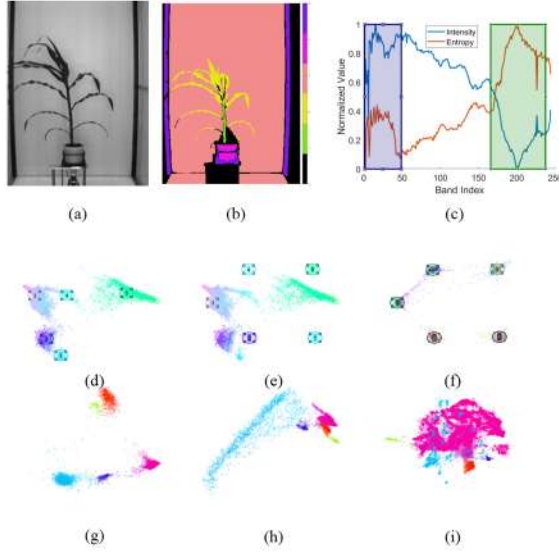
ure 6(c). Five different substances are labeled in this dataset, including Stem, Leaves, Background, Pot, and Frame. In the inferred scatterplot in Figure 6(f), five clusters of points can be easily located, which correspond very well to the five different substances in the groundtruth. The spatial relationship between the real objects can be inferred from the scatterplot. In Figure 6(f), cluster one connects to cluster two, cluster two connects to cluster three, and so on. In the original images, the stem is in contact with the leaves, and the leaves are in contact with the background. However, as the leaves are not in contact with the frame, there is no connection between cluster two and cluster five. Figure 6(g)(h)(i) are the results generated using ICA, LDA, and t-SNE, respectively. Compared to the result from these traditional methods, our inferred scatterplot not only clearly separates different substances but also preserves their relationship.

Figure 7 shows the result of a maize plant. The five labels are the same as those of the sorghum plant. Similar to the result of the sorghum plant, the inferred scatterplot in Figure 7(f) can also show five clusters of points clearly and reveal the relationship among these clusters, compared to the traditional methods in Figure 7(g)(h)(i).

Our colleagues in agronomy and horticulture commend that our tool has facilitated them to identify different parts of the plant, and these results could not be obtained in the existing work of plant hyperspectral image analysis.

## Discussion

In this section, we compare the performance of results generated using different features for the neural network. We also show how we can transfer the network trained for one dataset to another dataset without re-training the network.



**Figure 7.** The result of a maize plant.

### Input Selection

In the existing methods based on neural networks, usually all hyperspectral images are directly fed into a neural network. In our work, we have chosen to use the derived hyperspectral features based on statistical metrics and derivatives for the input of the neural network. As there are several combinations to choose, it is worth discussing how these selected hyperspectral features would impact the generation of a scatterplot. Thus, we compute the Mean Square Error (MSE) of the scatterplot generated using an individual combination of hyperspectral features with the scatterplot generated using original hyperspectral images.

Table 2 shows the MSE of the inferred scatterplot with different inputs. Org. represents the original data. 1st and 2nd mean the first and second derivatives. HyperCurve means the original intensity data of hyperspectral images. HyperMetrics means all the statistical metrics. Thus, each entry in the table is the MSE value corresponding to a hyperspectral feature generated by a different combination. For example, the entry of 1st and HyperCurve corresponds to the first derivative of the original intensity data, and the entry of 2nd and HyperMetrics corresponds to all the statistical metrics applied on the second derivative of the original intensity data. The + operator means the concatenation of features. We can see that the MSE of the inferred scatterplot

generally decreases with more features used.

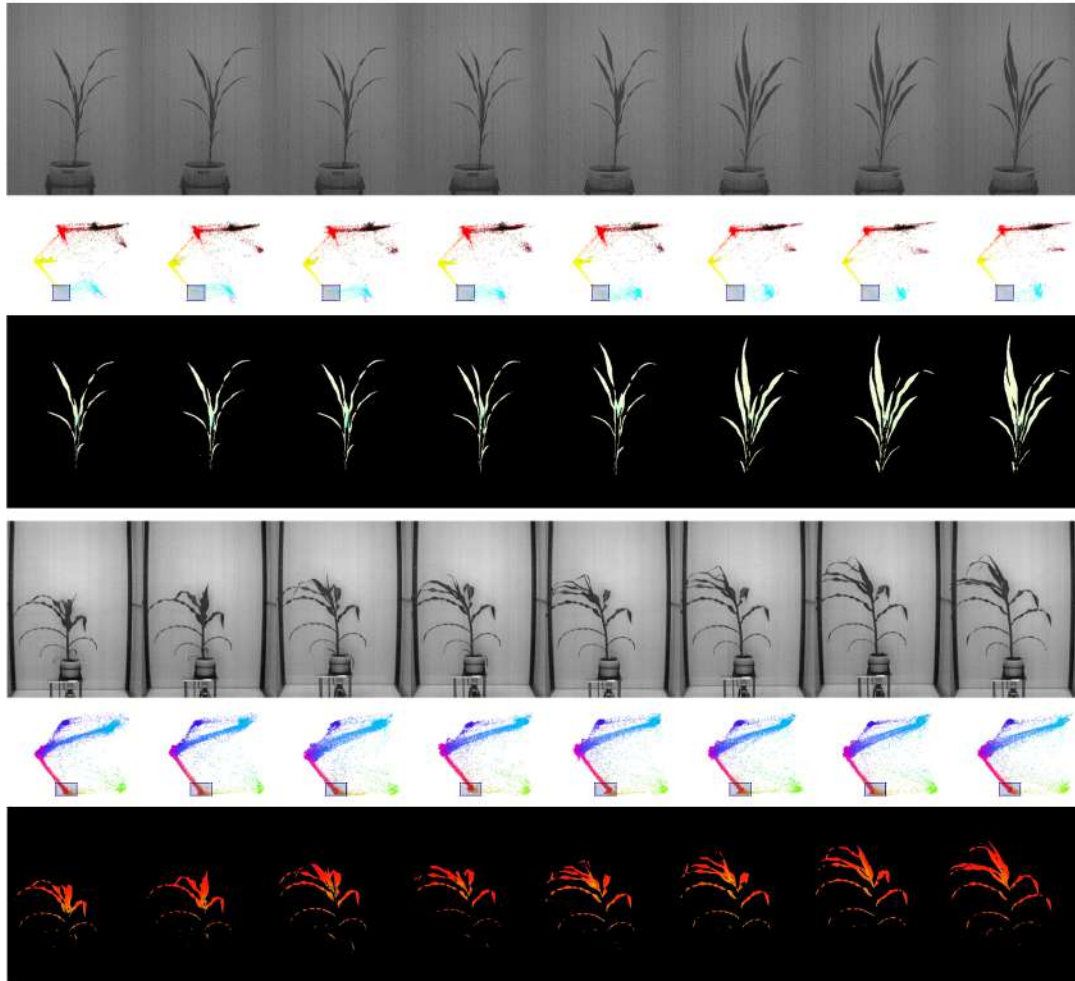
### Network Transfer

In our work, a neural network is trained to generate a scatterplot. If a set of hyperspectral images have similar spectral ranges and substances as another set, the network trained for one dataset can be directly used for another. We call this process network transfer.

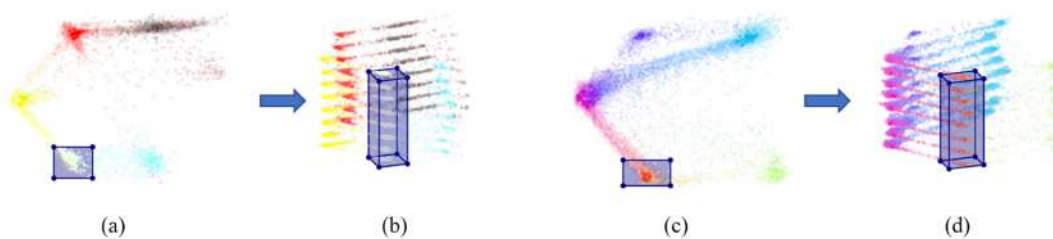
As shown in Figure 8, a sorghum plant and a maize plant are imaged each day over eight days. Then, we calculate the hyperspectral features using the same BOIs as in Figure 6(c) or Figure 7(c). Finally, we reuse the same neural network trained in Figure 6 or Figure 7 to infer the scatterplots for different sets of hyperspectral images. The first and fourth rows in Figure 8 are samples of the original hyperspectral images and the second and fifth rows in Figure 8 are the corresponding scatterplots. Here all the pixels are used to generate the scatterplot. The distribution of points in the scatterplots is slightly different from that in Figure 6(f) or Figure 7(f) as more points (most corresponding to pixels on the boundaries between objects) are included. It can be seen from these scatterplots that, although the sizes of the plants change over time, the positions of the clusters remain roughly the same in the scatterplots. Thus, this can facilitate a plant scientist to compare datasets from different time points. For example, if the scientist wants to examine how the leaves change over time, the clusters on the lower left corner can be selected from these scatterplots. For sorghum or maize, there is one blue rectangle at the same location in the scatterplots. If the points in the blue rectangles are selected, the corresponding leaves can be visualized, as shown in the third and sixth rows of Figure 8.

If the scatterplots are stacked together, a 3D scatterplot can be obtained. A 2D rectangle in the  $xy$  view of the stacked scatterplot corresponds to a 3D cuboid in the original stacked scatterplot. For example, the 2D selections in Figure 9(a)(c) correspond to the 3D selections in Figure 9(b)(d) that further correspond to the leaves of sorghum and maize in Figure 8, respectively. Our collaborators have commended that network transfer and batch selection greatly simplify the process of studying different parts of the plant in a time-varying situation, and provide an easier and more robust





**Figure 8.** The result of a sorghum (maize) plant over eight days. First (fourth) row: one of the original hyperspectral images of each day. Second (fifth) row: scatterplots generated using a pre-trained network. Third (sixth) row: selected leaves corresponding to the points in the blue rectangles in the scatterplots in the second (fifth) row. All the blue rectangles are located at exactly the same position.



**Figure 9.** Cluster selection in 3D stacked scatterplots. A 2D rectangle in the  $xy$  view of the stacked scatterplot corresponds to a 3D cuboid in the original stacked scatterplot. (a) and (c) are the  $xy$  views. (b) and (d) are the 3D views. (a) and (b) show the selection of sorghum leaves. (c) and (d) show the selection of maize leaves.

means to identify individual plant components and track their dynamics.

## Conclusion

Our solution keeps users in the analytical loop and facilitates them to interactively explore different substances from hyperspectral images. We also show that once a network is trained, it can be easily transferred to other datasets with similar hyperspectral contents. This can greatly help scientists study large sets of time-varying hyperspectral images with simple selections. We have demonstrated the effectiveness of our techniques with the examples in remote sensing and plant science. Our analytical pipeline and visual designs could be generalized and applied to other similar applications involving hyperspectral images.

We note that in an initial scatterplot, for the same substance, the points are usually close to each other and the colors are also similar, which provides a good indication for a domain expert to define cluster centers. If needed, the user can adjust cluster centers to improve image fusion results. While this is conducted in a trial-and-error manner, our domain experts usually can generate desired results with a few iterations, given that they have substantial knowledge of hyperspectral images and image fusion results, and also there are a limited number of substances. However, with an increasing number of substances and less experienced users, it can become difficult for users to manually identify and drag the circles to new positions. We are improving the scalability of our approach by exploiting automatic or semi-automatic methods (e.g. the force-directed method [11]) and facilitating users to select and separate the circles. We plan to continue this study and report corresponding findings in our future work. In addition, the features and neural networks are selected empirically in our current solution, and we would like to explore more hyperspectral features and more types of neural networks to increase the quality of the scatterplot.

## ACKNOWLEDGMENT

This work has been supported by the National Science Foundation through grants OIA-1736192, DBI-1564621, IIS-1652846, and IIS-1423487. The authors are grateful to the staff members at the University of Nebraska-Lincoln's Greenhouse

Innovation Center for their support in the data collection process.

## REFERENCES

1. C.-I. Chang. *Hyperspectral imaging: techniques for spectral detection and classification*, vol. 1. Springer Science & Business Media, 2003.
2. Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi. Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 54(10):6232–6251, 2016.
3. M. Cui, A. Razdan, J. Hu, and P. Wonka. Interactive hyperspectral image visualization using convex optimization. *IEEE Transactions on Geoscience and Remote Sensing*, 47(6):1673–1684, 2009.
4. B. M. Devassy and S. George. Dimensionality reduction and visualisation of hyperspectral ink data using t-SNE. *Forensic science international*, 311:110194, 2020.
5. X. Geng, L. Ji, and K. Sun. Principal skewness analysis: algorithm and its application for multispectral/hyperspectral images indexing. *IEEE Geoscience and Remote Sensing Letters*, 11(10):1821–1825, 2014.
6. G. Healey and D. Slater. Models and methods for automated material identification in hyperspectral imagery acquired under unknown illumination and atmospheric conditions. *IEEE Transactions on Geoscience and Remote Sensing*, 37(6):2706–2717, 1999.
7. M. Kawato, Y. Maeda, Y. Uno, and R. Suzuki. Trajectory formation of arm movement by cascade neural network model based on minimum torque-change criterion. *Biological cybernetics*, 62(4):275–288, 1990.
8. S. J. Kim, S. Zhuo, F. Deng, C.-W. Fu, and M. Brown. Interactive visualization of hyperspectral images of historical documents. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1441–1448, 2010.
9. S. Li, X. Kang, and J. Hu. Image fusion with guided filtering. *IEEE Transactions on Image Processing*, 22(7):2864–2875, 2013.
10. A. Meka and S. Chaudhuri. A technique for simultaneous visualization and segmentation of hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 53(4):1707–1717, 2015.
11. J. Tang, J. Liu, M. Zhang, and Q. Mei. Visualizing large-scale and high-dimensional data. In *Proceedings of the 25th international conference on world wide web*, pp. 287–297, 2016.
12. F. Zhu, Y. Pan, T. Gao, H. Walia, and H. Yu. Interactive visualization of time-varying hyperspectral plant images for high-throughput phenotyping. In *2019 IEEE Inter-*

**Feiyu Zhu** is a postdoctoral researcher at Sichuan University. He received a B.S. degree from the University of Electronic Science and Technology, China, an M.S. degree from Clemson University, and a Ph.D. degree in Computer Science from the University of Nebraska-Lincoln. He has been studying the visualization of hyperspectral images and plant phenotyping. This work was conducted as part of his Ph.D. study. Contact him at feiyu.zhu@huskers.unl.edu.

**Yu Pan** is a Ph.D. student in the Department of Computer Science and Engineering at the University of Nebraska-Lincoln. He works on scientific visualization with deep learning techniques. He obtained a B.S. degree from the University of Electronic Science and Technology, China, and an M.S. degree from the Illinois Institute of Technology. Contact him at yu@huskers.unl.edu.

**Tian Gao** is a Ph.D. student in the Department of Computer Science and Engineering at the University of Nebraska-Lincoln. His research interest is on plant phenotyping and phenotype data analytics. He obtained a B.S. degree from AnHui University, China, and an M.S. degree from the University of Science and Technology, China. Contact him at tgao@huskers.unl.edu.

**Harkamal Walia** is an associate professor in the Department of Agronomy and Horticulture at the University of Nebraska-Lincoln. His research focuses on crop abiotic stress tolerance, phenomics, and functional genomics. He obtained a B.S. degree in Plant Breeding and Genetics from Punjab Agricultural University, India, and a Ph.D. degree in Plant Biology from the University of California, Riverside. Contact him at hwalia2@unl.edu.

**Hongfeng Yu** is an associate professor in the Department of Computer Science and Engineering at the University of Nebraska-Lincoln. His research focuses on data analysis and visualization, and high-performance computing. He received a B.S. degree and an M.S. degree in Computer Science from Zhejiang University, China, and a Ph.D. degree in Computer Science from the University of California, Davis. Contact him at hfyu@unl.edu.