Machine Vision and Applications (2023) 34:100
https://doi.org/10.1007/s00138-023-01452-9

ORIGINAL PAPER

Explainable interactive projections of images

Huimin Han[1] · Rebecca Faust[1] · Brian Felipe Keith Norambuena[1,3] · Jiayue Lin[1] · Song Li[2] · Chris North[1]

Received: 18 March 2023 / Revised: 15 July 2023 / Accepted: 15 August 2023 / Published online: 13 September 2023

Abstract
Dimension reductions (DR) help people make sense of image collections by organizing images in the 2D space based on similarities. However, they provide little support for explaining why images were placed together or apart in the 2D space. Additionally, they do not provide support for modifying and updating the 2D representation to explore new relationships and organizations of images. To address these problems, we present an interactive DR method for images that uses visual features extracted by a deep neural network to project the images into 2D space and provides visual explanations of image features that contributed to the 2D location. In addition, it allows people to directly manipulate the 2D projection space to define alternative relationships and explore subsequent projections of the images. With an iterative cycle of semantic interaction and explainable-AI feedback, people can explore complex visual relationships in image data. Our approach to human–AI interaction integrates visual knowledge from both human-mental models and pre-trained deep neural models to explore image data. We demonstrate our method through examples with collaborators in agricultural science and other applications. Additionally, we present a quantitative evaluation that assesses how well our method captures and incorporates feedback.

Keywords Interactive dimension reduction · Semantic interaction · Explainable AI · Image data

1 Introduction

People commonly use dimension reduction (DR) methods to explore data for sensemaking tasks [1]. DR methods excel at mapping high-dimensional data to a low-dimensional space (typically 2D) while preserving meaningful structure and relationships. Several methods add interaction to enable exploration, modification, and understanding of the 2D space. For example, some systems incorporate semantic interactions which couple cognitive and computational processes by inferring meaning behind interactions and updating the model accordingly [2].

However, most interactive DR methods have limited support for image data, often representing images as arrays of pixels and treating them the same as tabular data. This not only limits the DR's ability to determine similarities between images but also often inhibits interaction methods for understanding the 2D space. For example, Self et al.'s Andromeda uses Weighted Multidimensional Scaling (WMDS) to create an interactive DR that supports semantic interaction for exploring and understanding 2D projection spaces via model steering [3]. After an interaction, the model learns new weights on the input dimensions that infer meaning from the interaction and explain the information learned by the projection. However, when a dataset does not have interpretable dimensions, these explanations become meaningless. What's more, because a single pixel has an arbitrary meaning across all images, weighting the same pixel in each image does not have a uniform effect on all of the images.

✉ Rebecca Faust
rfaust@vt.edu

Huimin Han
huimin@vt.edu

Brian Felipe Keith Norambuena
brian.keith@ucn.cl

Jiayue Lin
jiayuelin@vt.edu

Song Li
songli@vt.edu

Chris North
north@vt.edu

[1] Computer Science, Virginia Tech, Blacksburg, VA 24061, USA

[2] School of Plant and Environmental Sciences, Virginia Tech, Blacksburg, VA 24061, USA

[3] Department of Computing and Systems Engineering, Universidad Católica del Norte, 1249004 Antofagasta, Chile

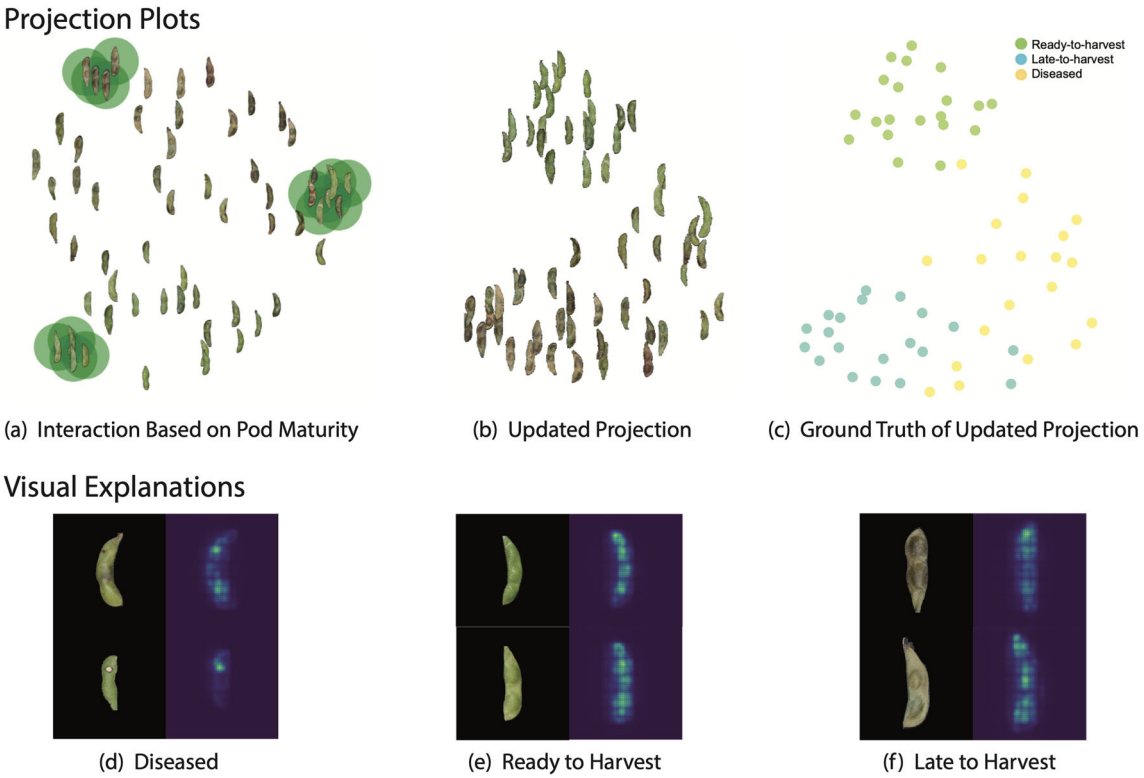


Fig. 1 Interactions to explore maturity level in edamame pod images. **a** shows user manipulations based on maturity level. **b** Shows the updated projection while **c** shows the ground truth maturity level. **d–f** Shows the explanations of important image features for each maturity level

Thus it does not make sense to directly project images from pixel arrays.

We know from past research that deep neural networks excel at extracting meaningful features from images and embedding them into a new representation [4]. Classifiers commonly use these embeddings, achieving high accuracy which indicates that the embeddings must be well suited for finding similarities between images. The question then remains, how can we use these feature embeddings to create more meaningful projections of image data and capture human feedback?

In this paper, we present an interactive DR method, built from Self et al.'s Andromeda, that supports semantic interaction for exploring projections of image data. Our method leverages the feature embeddings extracted from a convolutional neural network to project image data to a low-dimensional space using WMDS while supporting semantic interaction to enable people to explore and update the projection space. Our method enables people to directly manipulate the 2D locations of images to define new pairwise relationships in the 2D space. Based on the changes induced by these manipulations, the method learns new projection weights that best respect the user-defined relationships. Using these weights to re-project the images, people can observe the impact of those relationships on the projection space. Each dimension now represents some feature of the images, rather than an arbitrary pixel, but these dimensions are still not directly interpretable. Increasing the weight of a feature increases its importance in the projection but still does not provide any insight into the information learned. Thus, while updating the weights now has an inherent meaning, people have no real understanding of this meaning. That brings us to our second question: How can we translate the learned weights back to the image space?

In addition to providing an interactive DR, our approach provides explanations of features of importance in the 2D space through the use of a weighted backpropagation algorithm. We adapt a traditional visual backpropagation method for generating saliency maps [5] to apply the feature weights from the projection. Doing so creates saliency maps that highlight the image features learned from the semantic interaction. Thus, we are able to push the information learned from the interaction back through the network to the image space, where people can interpret it.

Our method helps people explore multiple projections of their image data through semantic interactions and explain the effects of these interactions on the placement of images through saliency maps. Figure 1 presents an example using our method, with a full description in Sec. 5.1. We note that our motivations for investigating 2D projection spaces stem

from our desire to support human cognition and sensemaking when working with image collections through dimensionality reduction. In particular, we note that organizing images in a 2D space based on similarities corresponds well with how people naturally think and reason about visual information. In contrast to regression models, our focus is not on predicting exact numerical values or fitting data to a particular model, but rather on capturing and visualizing the inherent relationships and structures within image data. Additionally, the objective of 2D DR is not to perform clustering, as is the case with interactive clustering methods, but rather to reveal the continuous and frequently complex relationships that exist between images. By representing images in a 2D plot where distance reflects similarity, we provide an intuitive and approachable framework for individuals to explore and interpret diverse visual data.

This paper extends the contributions of [6] to expand the evaluation with new usage scenarios and a quantitative evaluation. The extended contributions of this paper include:

- An interactive-AI method for dimension reduction that semi-automatically projects images based on visual knowledge from both pre-trained neural models and human feedback.
- An explainable-AI method for saliency mapping through weighted backpropagation that explains important image features.
- Four usage scenarios that illustrate real-world examples of image exploration tasks supported by our method.
- A quantitative evaluation demonstrating our methods' ability to organize image data according to interactive feedback.

2 Related work

Our work draws elements from interactive dimensionality reduction techniques, semantic interaction methods, and explainability in deep learning. In this section, we start by discussing related works from the interactive dimensionality reduction literature. Next, we focus on semantic interaction and its applications in sensemaking. Finally, we discuss explainability techniques for deep-learning methods in the context of image data.

2.1 Interactive dimensionality reduction

Dimensionality reduction techniques are commonly employed to analyze and visualize high-dimensional data by projecting it onto a 2D or 3D space [7]. Alone, DR algorithms typically produce a static projection space with no means for exploration or manipulation. Thus, many scholars sought to develop *interactive* DR techniques capable of capturing user feedback and subsequently modifying the projection.

Some interactive DR methods create a bi-directional workflow where people can alter data in the high-dimensional space to see the effect on the 2D location and vice versa [8, 9]. Other works explore the idea of backward (or inverse) projections that allow people to select locations in the 2D space and generate corresponding high-dimensional representations [10, 11]. Eler et al.'s work specifically targets image data, providing interactions for exploratory tasks, such as zooming into specific projection regions, displacing points to resolve overlapping, and displaying the nearest neighbors of selected images [12].

Many works exist on interactively steering projections. Several take the approach of requiring people to define control and organize control points, which are then used to project a larger collection of data while maintaining local structures around control points [13–15]. Others learn new distance functions for MDS to update the projection to best respect user manipulations [3, 16]. Fujiwara et al. provide a visual analytics framework for comparative analysis, providing interactions to manipulate and update projections to illustrate the similarities and differences between clusters of points [17].

Our work expands on past work by specifically targeting imaged data to provide both projection-steering interactions and visual explanations of the 2D space. We extend Self et al.'s Andromeda [3]. Andromeda allows people to directly manipulate the 2D location of data points and updates the projection model to incorporate human feedback into the projection. We propose an extension to Andromeda that supports image data via deep-learning feature representations and provides visual explanations of the important image features, before and after human feedback.

2.2 Semantic interaction

Semantic interactions exploit the natural interactions in visualizations to learn the intent of the user and then, based on these interactions, update the underlying model and its parameters [18]. In the context of sensemaking, semantic interactions capture the analytical reasoning of the users [19], and support analysts throughout the sensemaking process [20].

Most semantic interaction systems work using a dimensionality reduction model, similar to the interactive dimensionality reduction methods described in the previous section. Semantic interaction is a bi-directional pipeline [21] and requires capturing the changes in the visualization and turning them into changes to the model. In the dimensionality reduction case, this is usually done through the use of an inverse transformation (e.g., inverse WMDS) [22]. There are several models that can be used to solve the bi-directional

transforms required to implement semantic interactions, such as Observation-Level Interaction [23], Bayesian Visual Analytics [24], and Visual to Parametric Interaction [25].

Previous work has also shown how to integrate deep-learning models with semantic interaction techniques. For example, Krokos et al. [26] designed an interactive tool to help humans label data points for semi-supervised learning using a deep-learning model. Bian and North [27] developed a semantic interaction model for text analytics integrating traditional dimensionality reduction techniques with a neural network as its core component. Bian et al. [28] continued the development of these semantic interaction models and designed an explainable AI framework based on counterfactuals that help users understand the generated projection.

2.3 Explainability in deep learning

Scholars have proposed several explainability methods for convolutional neural network (CNN) models, the backbone of most image-based deep-learning applications. Bojarski et al. [5] proposed a visualization method that shows which pixels of an input image contribute the most toward the predictions of a CNN model. In particular, their technique allows debugging CNN-based systems by highlighting the regions of the input image that have the highest influence on the output of the model. Zeiler and Fergus [29] developed a novel visualization technique that provides insight into the intermediate feature layers of a CNN in a classification task. Zhou et al. [30] use a global average pooling layer to shed light on how this layer enables CNN models to localize objects in images. In particular, their approach generates a Class Activation Map (CAM) using global pooling. However, while these explanation techniques are powerful, they are designed for specific CNN-based models. To address this weakness, researchers have proposed visual explanation techniques for a large class of CNN-based models. For example, Selvaraju et al. [31] generated CAMs based on gradient information of target concepts (Grad-CAM). Grad-CAM provides fine-grained explanations of the CNN predictions but suffers from performance issues with multiple occurrences and single-object images.

Despite the recent advances in explainable deep learning for image data, there is a dearth of studies exploiting explainable deep-learning techniques for interactive DR in the context of image analysis. Thus, our work seeks to fill this gap and combine interactive DR for images with explainable deep-learning techniques. In particular, we base our work on the method of Bojarski et al. [5], as visual backpropagation provides an efficient way to generate explanations of relevant image features for the users by pushing the weights obtained in the interactive DR loop through the backpropagation process.

3 Tasks

Before discussing the details of our method, we first must discuss the sensemaking tasks of someone using our tool. Pirolli and Card described the sensemaking process as having two primary loops: the foraging loop and the sensemaking loop [32]. The foraging loop focuses on searching and filtering information and extracting evidence. The sensemaking loop then uses this information to iteratively construct representational schemas as well as generate and test hypotheses about the data.

In the context of image data, simply looking at every image does not provide sufficient information to make sense of the data. The foraging loop requires filtering and extracting sets of images relevant to the task at hand. Then, those images must be organized into a schema that provides a structured representation for consuming the image data and testing hypotheses. The process of generating and refining the schema typically requires several iterations of foraging for information under the current schema, updating the schema based on the new information, and evaluating how the schema fits the task at hand to determine if it requires further refinement.

Our method supports this schematization step through iterative exploration of the images and refinement of the 2D representation to reflect prior knowledge of the analysis task. Through discussions with collaborators in the plant sciences, we identified the following tasks to support this iterative process: (1) define custom similarities based on prior knowledge and (2) link human- and machine-defined similarities

These tasks create a synergy between the machine and the human where they work together as a team, teaching each other what they have independently learned from the data. In the end, we create an analysis pipeline where the human perceives the data, conveys their knowledge to the machine, and the machine then re-organizes the data based on this information, while providing explanations of its reasoning. The remainder of this section discusses these tasks in greater detail.

3.1 Define custom similarities based on prior knowledge

When analyzing data, people typically have some prior knowledge about the data, such as what categories of or similarities between images they expect to exist within the data. For example, in a set of edamame pod images, the analyst may expect images of healthy pods and diseased pods. Static dimension reduction plots, may or may not adequately reflect this prior knowledge. In the previous example, the person analyzing may want to inspect healthy vs diseased pods, but the model may not naturally recognize these differences. Furthermore, static projections do not enable people to

explore different projections defined under different guidelines. To enable hypothesis testing, people must be able to steer the projection to define similarities in the data in a way that reflects their prior knowledge. With our method, people directly manipulate the 2D location of images to define new relationships within the data that the model then learns and uses to re-project the images accordingly.

3.2 Link human-defined and machine-defined similarities

The previous task focuses on teaching the projection model to incorporate human knowledge. However, while it helps the model learn human knowledge, it does not help people understand the model's knowledge. People need ways to inspect the image features most important to the 2D projection. This helps them not only understand the 2D space but also validate the model's perception of their interaction and potentially identify other image similarities/differences beyond the knowledge they intended to teach the model. Our method provides saliency maps that illustrate the features of the image that the projection most heavily used to place the image. Viewing the explanations of multiple images provides insight into why the model placed them near or far from each other and provides a means for understanding the 2D space.

4 Workflow and methodology

In this section, we describe the expected user workflow and interactions, as well as the underlying methodology.[1] Figure 2 gives an overview of the workflow while Fig. 1 presents an example of using this workflow.

4.1 Initial state

Upon loading the data, our method extracts the neural embeddings of the images to project them into the 2D space. It then uses Weighted Multidimensional Scaling (WMDS) to project the features into 2D. For the initial projection, our method assumes no prior information and thus treats all features in the neural embedding with equal importance. The resulting plot provides the initial view into the similarities of the data and serves as the starting point for the exploratory analysis. We chose WMDS because it uses pairwise similarities as the input for projection and thus changes in the 2D similarities conceptually map directly back to the input space.

Feature extraction Feature extraction is an important technique in computer vision widely used for tasks such as object detection and image classification [33]. Existing feature-extraction methods for image data include traditional approaches such as Harris Corner Detection [34] and Scale-Invariant Feature Transform (SIFT) [35]. Recently, deep-learning models have become popular for feature extraction in images [36]. In particular, Convolutional Neural Networks (CNN) have shown great power in image-related tasks [37]. Thus, using CNNs has become the standard in feature extraction [38].

Furthermore, the rise of transfer learning enables researchers to utilize the power of pre-trained models instead of training a deep neural network from scratch [39]. Our method uses the pre-trained ResNet18 [40] as a fixed feature extractor to generate feature vectors from images.

Given an image dataset $\mathcal{D}$, we forward propagate the images through the network with the fully connected layer removed. The final representations are denoted as:

$$\mathcal{X} = ResNet_{\text{pre-trained}}(\mathcal{D}) \tag{1}$$

The feature space $\mathcal{X}$ is a 512-dimensional space used to represent the images. Each x_i is the output of applying average pooling to the final feature map of the network. We use $\mathcal{X}$ as the input to the interactive dimension reduction loop.

Weighted Multidimensional Scaling Using the extracted image features ($\mathcal{X}$) as input, we perform MDS on a weighted data space (WMDS) to project the images to 2D, using the following function:

$$y = \underset{y_1,\ldots y_n}{\arg\min} \sqrt{\sum_{i<j\leq N} (d_L(y_i, y_j) - d_H(w, x_i, x_j))^2} \tag{2}$$

where N is the number of points in the dataset, $d_L(y_i, y_j)$ is the low-dimensional distance between y_i and y_j, and $d_H(w, x_i, x_j)$ is the weighted high-dimensional distance between the feature representations x_i and x_j, given the dimension weights w. We calculate d_H by first weighting the data space using w (i.e., $\mathcal{X} * w$) and then calculating the pairwise distances in the weighted data space. For the initial projection, we initialize w with equal weights for every dimension, relying solely on the raw image features to organize the images.

4.2 Interaction and inverse dimension reduction

From the initial state, people can directly manipulate the projection plot, dragging points into new positions in the 2D space (as shown in Fig. 1a). Dragging points to new positions defines new pairwise relationships to teach the projection model. For example, in Fig. 1a, an analyst projects a collection of edamame pods that contain three phenotypes of pods: ready to harvest, late to harvest, and diseased. However, the initial projection of our pods, the projection does not differentiate these phenotypes. By selecting and dragging a few

[1] The implementation of our method can be found at https://github.com/infovis-vt/Andromeda_IMG.

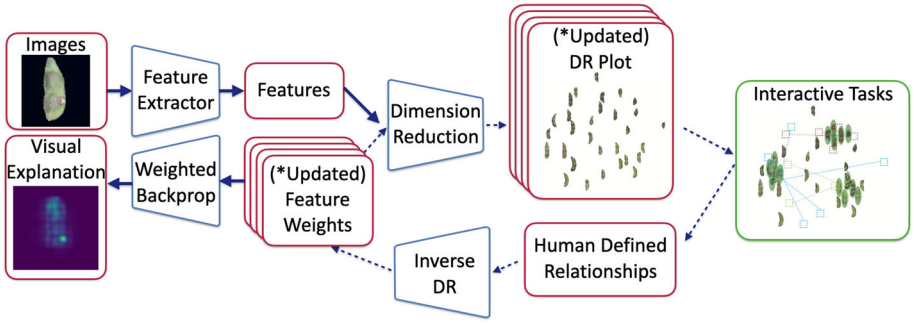


Fig. 2 An overview of our workflow. First, we extract image features using a deep-learning feature extractor which we then pass to an interactive DR method (WMDS) that facilitates semantic interactions. The dark blue dotted arrows signify the human interaction loop. After interactions, we pass the newly defined relationships to the inverse DR where it learns new projection parameters (updated feature weights) that best respect them. These feature weights are used to re-project the images using the DR (generating an updated DR plot) and start the interaction loop over again. The interaction loop can be repeated many times

representative images of each type to opposing positions, the analyst indicates to the machine that those images are dissimilar and should be organized accordingly. Once the analyst completes their interaction, our method uses these relationships to optimize the projection weights to create a projection layout that best respects the defined relationships.

Inverse dimension reduction To facilitate interactive dimension reduction, we use inverse WMDS (WMDS^{-1}) to update the projection after semantic interactions, as originally described in Andromeda [3].

After the analyst re-positions a subset of the points, y^*, we perform WMDS^{-1} to calculate new weights optimal for maintaining the specified relationships, thus capturing human feedback. WMDS^{-1} uses the following equation to update the weights:

$$w = \underset{w_1,\ldots w_d}{\arg\min} \sqrt{\frac{(\sum_{i<j\leq N} (d_L(y_i^*, y_j^*) - d_H(w, x_i, x_j))^2}{\sum_{i<j\leq N} d_H(w, x_i, x_j)^2}} \quad (3)$$

This equation produces a vector of dimension weights that best respects the 2D pairwise similarities specified through the interactions. We normalize the weight vector to sum to 1, so as to normalize the HD distances to a roughly constant-sized space. We then re-project the images using equation 2 with the updated weights to create a layout that incorporates the analyst's feedback.

4.3 Visual explanations

To fully enable interactive projections, we must also enable people to inspect the information learned from their interaction. Our method provides visual explanations in the form of saliency maps to provide visual feedback and explanations of the information learned by the projection. The saliency maps highlight the important features in a given image, shown in Fig. 1d–f, such that the brighter pixels correspond to features of greater importance.

In the initial view, before semantic interactions, these explanations indicate the features of importance identified by the feature extractor that the projection model then uses to place the images. After an interaction, the optimized feature weights are pushed backwards through the feature extractor, using weighted backpropagation (described below), to generate new saliency maps that emphasize the features learned by the projection model. By inspecting the differences between the original saliency map and the post-interaction map, people can understand what features the projection learned from their interaction. This feedback enables people to better complete their tasks and refine their sensemaking schemas.

Weighted visual backpropagation Figure 3 illustrates our weighted visual backpropagation method. We base our proposed method on the visual backpropagation method proposed by Bojarski et al. [5]. This method computes the actual contribution of neurons to the feature representation, making the backpropagation fast and efficient. We make this method projection-aware by applying the projection weights to the backpropagation.

To implement our method, we utilize the feature maps after each ReLU layer. For the feature map of the last convolutional layer, we conduct channel-wise multiplication with the weights w obtained from the interactive DR loop to backpropagate the user's intent. We then average the other feature maps to get a single feature map per layer. The deepest single feature map, highlighted in green in Fig. 3, is deconvolved with the same filter size and stride as the convolutional layer immediately preceding it. This scales the feature map to match the size of the map in the previous layer. Then we pointwise multiply the deconvolved feature map by the averaged single feature map of the previous layer. This process is repeated until we reach the input image.

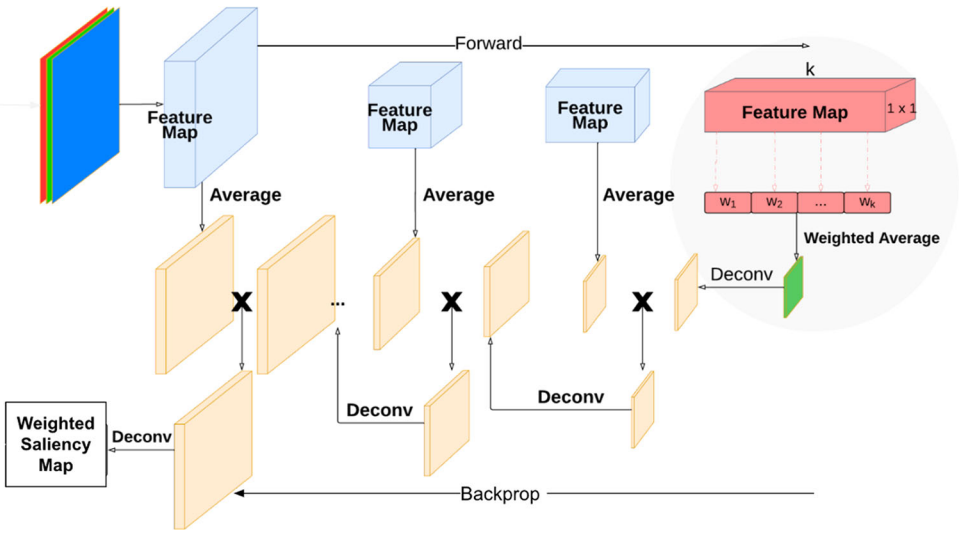


Fig. 3 Weighted visual backpropagation process

We keep our notation consistent with Bojarski et al. [5]. Note, we will only describe our modification to their method. For full details, please refer to Bojarski et al. Consider a convolutional neural network $\mathcal{N}$ with n convolutional layers. Let $\gamma(i)$ denote the value of pixel i of the input image and v represent a neuron. e represents an edge from some other neuron v' to v and a_e denotes the activation of v ($a_e = a(v)$). $\mathcal{P}$ denotes a family of paths. The contribution of the input pixel i, calculated by the original Visual Backpropagation method, is defined as:

$$\theta^{\mathcal{N}}_{VBP}(i) = c * \gamma(i) \sum_{P \in \mathcal{P}} \prod_{e \in P} a_e \tag{4}$$

For our method, we enable users to adjust the weights for the final network embeddings, which is the feature map of the last convolutional layer. To back-propagate the weighted feature map, we conduct channel-wise multiplication for the last feature map with weights gained from the interactive DR loop. We denote e_t as the edge that connects nodes from layer $(t-1)$ to layer t. Let k denote the kernels for each layer. The contribution of the input pixel i calculated by our Weighted Visual Backpropagation method is defined as

$$\theta^{\mathcal{N}}_{\text{WVBP}}(i) = c * \gamma(i) \sum_{P \in \mathcal{P}} \prod_{e \in P} a_{e_t} \tag{5}$$

where

$$a_{e_t} = \begin{cases} a(v) & \text{if } t \neq n, \\ a(v) * w_k & \text{if } t = n. \end{cases}$$

and w_k is the weight from the inverse projection corresponding to channel k of the feature map in the final layer.

5 Usage scenarios

In this section, we present two real-world usage scenarios to illustrate the utility of our method on image sorting tasks.

5.1 Edamame pods

We developed this usage scenario with collaborators in the plant sciences department [41]. Our collaborators identified the need for incorporating human perception into model development for identifying plant features. One use case of this idea stems from sorting images of edamame pods. Initially, they wanted to organize images of edamame pods based on maturity level. However, when exploring the images they also discovered that the pods contained varying numbers of seeds, which often correlates to the consumers' perception of quality. They envisioned that a method like ours would help them re-organize the images based on this newly identified feature and allow them to reuse the original model. In the remainder of this section, we discuss two scenarios for organizing images of edamame pods. For our example, we use a subset of their edamame pod dataset containing 60 images, with 20 images per maturity stage.

5.1.1 Maturity stage

The maturity stage of each pod is defined as either diseased, late-to-harvest, or ready-to-harvest. Here, we test how well our method can organize the images according to these phenotypes from human feedback and whether the features captured by the model to separate the images relate to the underlying phenotypes, illustrated in Fig. 1. First, we project the edamame pods to 2D. Then, we observe the visual phenotypes for maturity and interactively drag a subset of pods (highlighted in green) in order to group them

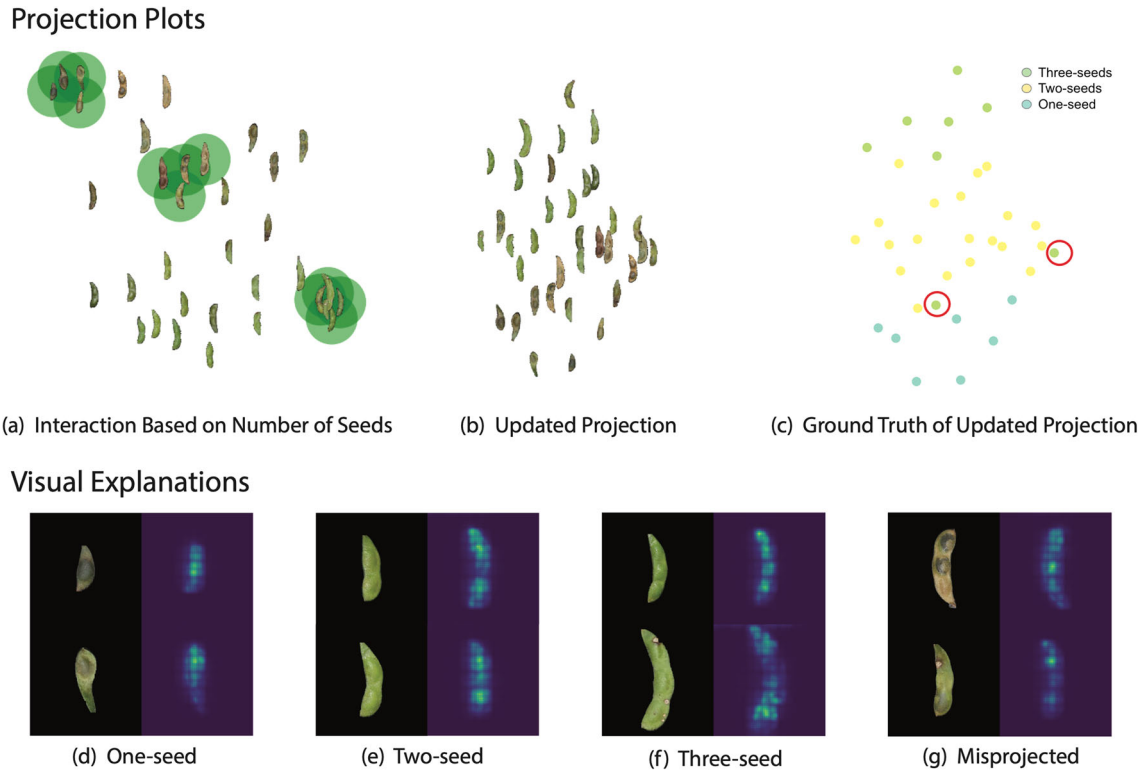


Fig. 4 Interactions to explore images based on the number of seeds. **a** shows the interaction based on seed count. **b** Shows the updated projection while **c** shows the ground truth seed count. **d–f** Shows the explanations of important image features for each seed count while **g** shows the explanations of two mis-projected images

into three clusters according to the desired phenotype categories, shown in Fig. 1a. We hypothesized that, through this interaction, the underlying model will learn new weights for the feature space that satisfy the newly defined projection and properly capture the user's mental model of pod maturity.

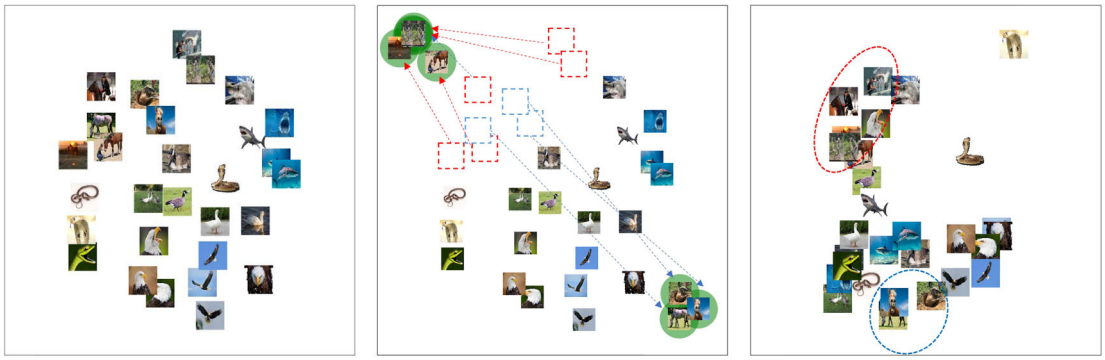
Figure 1b shows the updated projection (generated after approximately 25 s), which produced three main clusters of pods according to their maturity stage. Figure 1c shows the ground truth of the images. This indicates that the desired phenotypes were effectively captured by the weighted features and represented in the updated model.

The explainable feature visualizations of specific pods depict the most important visual features learned by the interactive model. In Fig. 1d we see that one of the important visual features learned by the model to determine the disease phenotype is a salient discolored spot. Similarly, in Fig. 1e, f, the model focuses on image areas correlated to important features of each pod. This provides insight into that parts of the pod are important for visually discerning the maturity stage. Furthermore, these results provide a link between human perception and machine learning.

5.1.2 Number of seeds

For the same pod dataset, we also want to explore a different visual phenotype: the number of seeds per pod. However, the images were not originally collected to determine the number of seeds. Thus, the number of seeds is a novel visual feature that can be observed directly by the end users but is not initially used to cluster images in the default projection. As before, the images of edamame pods are displayed in the 2D plot. We then interactively drag pods (highlighted in green) to group them into three clusters according to the number of seeds (one, two, or three), as shown in Fig. 4a. We hypothesize that by dragging a subset of the images, the underlying model will learn the weights for the feature spaces that satisfy the user-defined projection based on the number of seeds.

Figure 4b shows the updated projection. We find that the projection model captures the "number of seeds" phenotype. Figure 4c shows the ground truth of the updated projection, instead of well-separated groups, the updated projection shows a linear relationship. We notice that there are two "three-seed" pods projected closer to the "two-seed" pods. To learn more about why these two pods are mis-projected, we explore the visual feature explanations for each group.



(a) Initial Projection (b) User Manipulation (c) Updated Projection

Image Before After Image Before After

(d) Explanations for two images with both horses and humans, before and after the interactions.

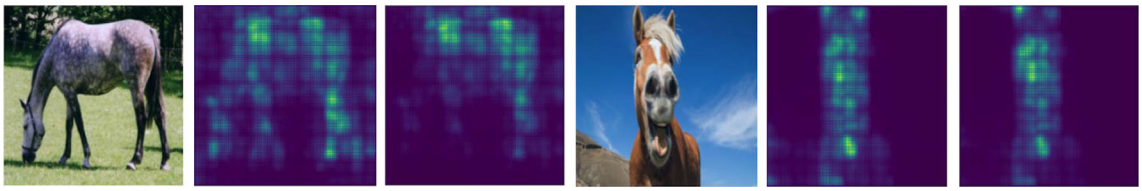


Image Before After Image Before After

(e) Explanations for two images with only horses, before and after the interactions.

Fig. 5 Usage scenario on the animals dataset: **a**, **b**, **c** show the process for exploratory analysis on a small subset of images. In **b** the user drags the "human and horse" images apart from the "horse" images to emphasize the "human" object. In the updated projection **c** the animals are projected near the bottom and images containing "humans" are clustered at the top (circled in red). **d** shows the saliency maps before and after the interactions for two images with humans and horses. The "after" saliency maps show greater levels of attention on the "human" object. In contrast, **e** shows the saliency maps for two images with only horses where the horse objects remain emphasized after the interaction

Figure 4d, e, f shows the saliency map for the three groups accordingly. We find that the most important CNN features mainly capture the overall shape of the pod, as well as the position and the "raised" area of the seeds to differentiate pods with different numbers of seeds. Yet for those two misprojected pods, they are either dominated by the disease spot or do not have the obvious shape of three seeded pods, as shown by Fig. 4g.

5.1.3 Open-mouth animals

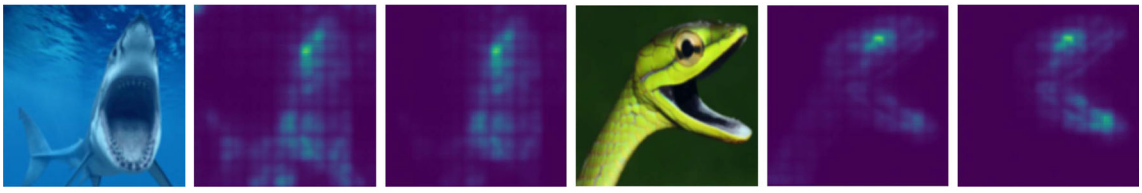
5.2 Animals

In this scenario, we use a dataset of images of animals from Kaggle [42]. This dataset consists of 5400 animal images in 90 different classes. For the task, we sampled a subset of this data, using only five classes of animals—horse, goose, shark, snake, and eagle—with five or six images per class. Figure 5 illustrates this usage scenario.

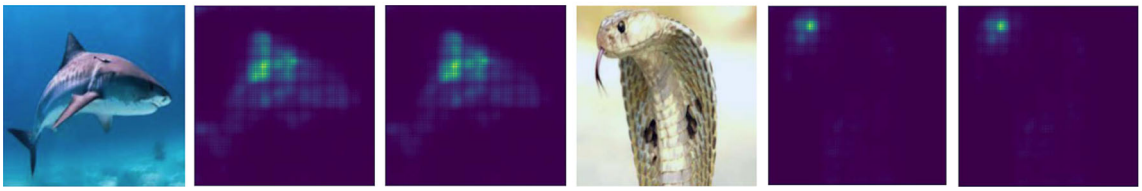
5.2.1 Human objects

After loading the data, our method creates the initial projection of the images, shown in Fig. 5a. The initial projection organizes the images such that animals of the same class are placed close together. However, after inspecting the projection we notice that some of the images contain both humans and animals. After this realization, we decide we want to inspect images of animals and humans separately from images only containing animals. We want to teach the underlying model to capture the concept of "human" rather than just grouping the images based on the animals. To do

(a) Initial Projection (b) User Manipulation (c) Updated Projection



(d) Explanations for an open-mouthed shark and snake.



(e) Explanations for a close-mouthed shark and snake.

Fig. 6 Usage scenario on the animals dataset: **a**, **b**, **c** show the process for exploratory analysis on a small subset of images. In **b** the user drags the "animal has mouth open" images apart from the "animal has mouth closed" images in the same animal category to emphasize the "mouth" object. In the updated projection **c** the animals that have their mouths open are clustered at the top (circled in red). **d** Shows the saliency maps before and after the interactions for an open-mouthed shark and snake. The "after" saliency maps show a greater emphasis on the snakes' mouths and a reduced emphasis on the sharks' bodies (focusing on the open mouth), thus capturing the "open mouth" feature. **e** Shows the saliency maps for a close-mouthed shark and snake, with the attention largely unchanged by the interaction

so, we drag the "human and horse" images apart from the "single horse" images as shown in Fig. 5b. After this, the underlying model learns the current user-defined layout and updates the entire projection based on the learned weights. Figure 5c shows the updated projection. In this projection, the images containing humans are projected together, while all the other animal images are re-projected accordingly, with animals of the same still projected in close proximity. Thus, all the pure animal images are separated from the images containing humans.

After teaching the projection to organize the images based on whether they contain a human, we want to inspect what features the projection used to place images and if the projection actually picked up on the human features in the image. Visual explanation method and inspect the saliency maps are used before and after the update, shown in Fig. 5d, e. To illustrate this, we selected two of the images containing humans and horses, shown in Fig. 5d. Before the user manipulates images, the underlying model projected images mainly based on animal content in the images as shown in the "Before" maps of Fig. 5d. Thus, the horse images are closer to each other in the projection space, as the machine mostly focuses on the horse object in the images. After the user manipulates the projection, the machine-learning model puts more attention on the humans as shown in the "after" maps of Fig. 5d. To compare, we inspect the explanations for two horse images that do not contain humans, shown in Fig. 5e. We see that, while the emphasis changes somewhat, it still focuses on the entire horse object. Using the visual explanations, we clearly see that the projection adequately inferred the meaning behind the interactions.

For the same animal dataset, we also found another visual feature to explore: whether the animal's mouth is open or not. Some of the animals in the images have their mouths open and

we want the projection to re-organize the images to separate the open-mouthed animals from the others. To convey this information, we drag a select set of open-mouthed animals apart from close-mouthed animals. For example, as shown in Fig. 6b, we pick two images from the snake, shark, and eagle groups. For each group, one of the images has the mouth open and the other has the mouth closed. The original projection, shown in Fig. 6a, organizes images so that the animals of the same type are projected together. We expect that after learning and re-projection, the open-mouthed animals and close-mouthed animals will be apart from each other and the model will increase the attention on the animal's mouths rather than the entire animal. Figure 6c shows the updated projection and we can see that all open-mouthed animals are grouped together, two new open-mouth sharks that we did not select to drag during the learning phase are also projected close to the other open-mouth animals, indicating that the projection learned our intent.

To verify that the model actually learned the intended information, we generate the saliency maps before and after the update to inspect the learned features, shown in 6d, e. We select two images of open-mouthed animals to inspect, one that we used in our interaction and one that the projection identified from our interaction, shown in 6d. Before the interaction, the underlying model projected images based on the entire animal in the images (the "before" maps in 6d). After the interaction, the model puts less attention on the shark's body and more attention on the snake's mouth as shown in the "after" maps in 6d, indicating the projection learned our intent and identified the intended features during the learning process. To compare, we also inspect the explanations for a close-mouthed shark and snake, as shown in Fig. 6e. We see that the interaction largely does not change the emphasized features of the close-mouthed animals.

6 Quantitative analysis

In addition to the use cases shown before, we perform a quantitative analysis to assess our method's ability to organize the images based on human feedback and evaluate the number of interactions necessary to produce a desirable organization. Ultimately, our system aims to steer projections based on analysts' prior knowledge. To evaluate our method's effectiveness at incorporating human feedback, we focus on the natural task of guiding the projection to separate images by distinct classes. We evaluate our method on two versions of this task: (1) organizing images by a distinct visual feature from a random projection and (2) shifting the projection from a layout using based on one feature to a layout using a different feature. Additionally, we evaluate how many interactions per image class are necessary to reach a well-organized layout. To measure the quality of the layout, we calculate an adjusted Silhouette score [43] of the clusters in the resulting projection. The remainder of this section describes the details of the evaluation.

6.1 Method

Our experimental design stems from the simulation experiments in [27]. To evaluate our method, we create a simulation engine that simulates semantic interactions. The interactions organize a subset of the images such that images of the same class are placed close together and images of different classes are far apart. From this organization, we learn new projection weights, use those weights to organize the whole set of images, and then evaluate the clustering in the layout. We run this simulation many times, with varying numbers of simulated interactions per class to evaluate the number of interactions necessary to reach a well-clustered layout.

6.1.1 Data

In this experiment, we use two datasets: images of animals [42] and images of edamame pods. We use the animal images for the first task, organizing images based on a distinct visual feature from a random layout. This dataset contains 300 images with five types of animal (horse, goose, eagle, shark, snake), giving 60 images per category. Using this dataset, the simulated analyst aims to guide the projection to identify and separate the images by the type of animal in the image. We use the edamame pods for the second task, to evaluate how well our method can re-organize images by a second feature. The pods are labeled with two features: the number of seeds in the pod and the maturity of the pod. With this dataset, we initially organize the pods by the number of seeds in each pod and simulate interactions to re-organize them by their maturity, measuring the quality of the resulting projection.

6.1.2 Simulation engine

The simulation engine consists of two main components: the interaction simulator and the layout evaluator. The simulation process consists of the following steps: (1) project the images using WMDS to create an initial layout of the data, (2) use the interaction simulator to select a subset of size n from each class and fully organize them into clusters (Fig. 7a), (3) learn new weights using WMDS^{-}1 that respect the simulated interactions and project the whole dataset using those weights (Fig. 7b, c), and last, (4) use the layout evaluator to measure the performance of the resulting layout. Steps (1) and (3) are described above in Sect. 4, while steps (2) and (4) are discussed in more detail below. We repeat this process many times for different numbers of interactions per class (different values of n).

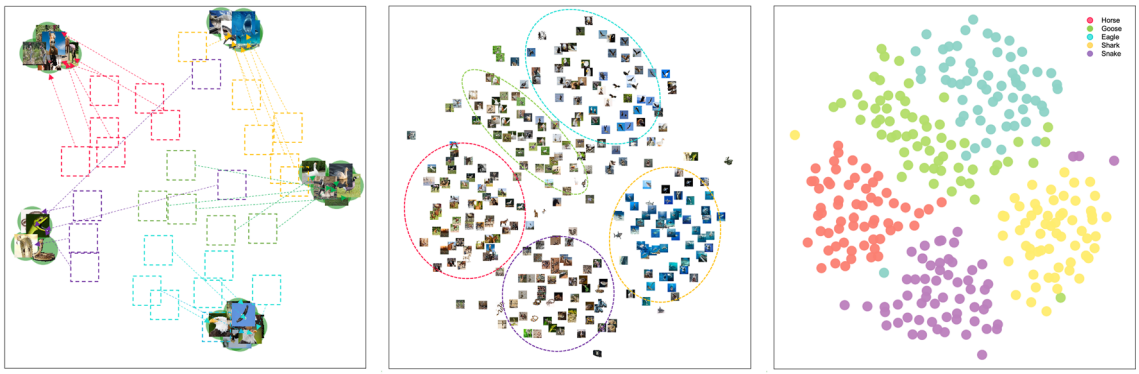


Fig. 7 Example of the simulation process. In **a** the analyst organizes a sample of images from each relevant label and our method learns new weights based on this layout. **b** Shows the projection of the full dataset using the learned weights, generalizing the layout based on the user's interactions. **c** Shows the performance of the resulting layout with respect to the ground truth of the dataset. The updated projection has a Silhouette score of 0.455

Interaction simulator For each semantic interaction, the simulator randomly selects n samples from each image class. It then generates the pairwise distance matrix using the following equation, where x_i and x_j are two of the randomly selected images:

$$||x_i - x_j|| = \begin{cases} 0 & \text{if } x_i \text{ and } x_j \text{ are from the same class} \\ \sqrt{2} & \text{otherwise} \end{cases}$$

With this equation, the simulated analyst places images of the same class directly on top of one another to show the model that they should be placed together. In contrast, it places images of different classes sufficiently far apart ($\sqrt{2}$) to teach the model that those images are dissimilar from one another. Figure 7a provides an example of the semantic interaction that the interaction simulator is mimicking. After simulating the interactions, the simulation engine uses our method to learn new weights that account for the relationships defined by the interactions and projects the entire dataset using these weights.

Layout evaluator To measure how effectively our method captures the simulated analyst's feedback, we calculate the adjusted Silhouette score of classes in the resulting projection [43]. The Silhouette score evaluates a clustering on two bases: cohesiveness and separation. The cohesiveness aims to minimize the separation within a given cluster while the separation aims to maximize the distance to the nearest clusters. It returns a value from -1 to 1, where values near zero indicate overlapping clusters, negative values indicate misassigned data and a positive score indicates the cohesiveness and separation of the clusters.

However, because our goal is to create a well-organized dimension reduction based on human feedback rather than a succinct clustering, we do not prioritize compact, highly separated clusters as valuable information may be contained in the spread of clusters. For example, in Sec. 5.1, the first phenotype that we teach the DR (maturity stage), the "diseased" phenotype spans both "ready-to-harvest" and "late-to-harvest" as well. Thus, while the pods have distinct classes, the spread of the clusters still contains useful information. s a result, the ideal Silhouette score would fall around 0.5, rather than 1. Conceptually, this would prioritize layouts where, on average, points are approximately twice as far from the points in the nearest class as they are from the points in their own class. To accommodate this, we multiply the Silhouette score by two, such that one becomes the ideal score for our DR, values less than one are too diffused, and values greater than one are too clustered. Figure 7c provides an example of a well-organized layout with an adjusted Silhouette score of 0.91.

6.2 Results

Task 1: Organize by distinct visual feature Figure 8 shows a plot of the Silhouette score against the number of points moved in each category. From this plot, we see that as we increase the number of points moved in each class our method steadily increases in its ability to organize the points. While the performance continues to increase, we see that after interacting with around five to ten points per category, the benefits of moving more points become marginal. Figure 7b, c shows an example layout after a user moves five points per class. We can see that by moving relatively few points from each class to define similarities in the dataset, our method creates a layout, with an adjusted Silhouette score of 0.91, that respects these relationships and effectively applies them to the greater dataset.

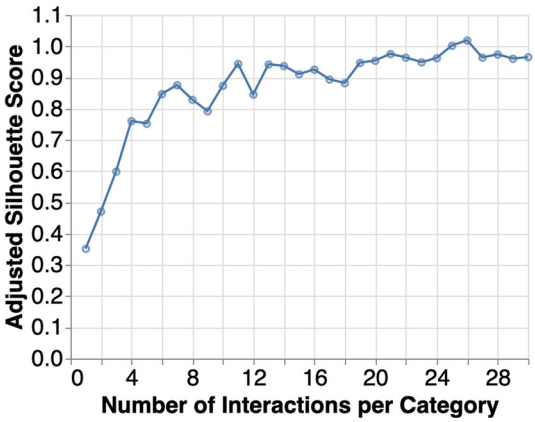


Fig. 8 The Silhouette score of the projection layout over the number of control points moved per category for the first task, organizing by a distinct visual feature

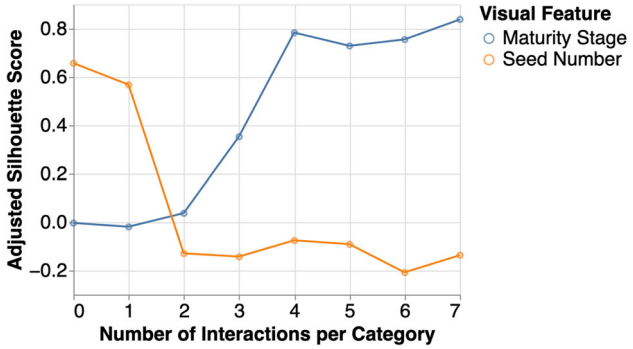


Fig. 9 The adjusted Silhouette score of the projection layout over the number of control points moved per category for the second task, re-organizing by a second visual feature

Task 2: Re-organize by a second visual feature Our second task focuses on re-organizing our DR layout based on a second distinct visual feature. For this example, we use the edamame pod dataset. We initially organize the pods based on the number of seeds in each pod. A natural follow-up task is to re-organize based on a different feature, namely the maturity stage of the pods. Thus, we evaluate how many interactions it takes to move between two DR layouts organized by different visual features. Figure 9 shows a plot of the adjusted Silhouette score for each visual feature against the number of points moved in each category. We see that our method quickly picks up on the new feature and begins adjusting the layout accordingly.

7 Discussion

General framework for analysis using deep-learning features One of the central problems with using deep-learning feature representations in data analysis is the loss of access to the original data features. Typically, people must sacrifice analysis transparency for performance. However, our method presents a framework in which we maintain access to the original data features by leveraging the underlying deep-learning model to create explanations from the underlying data features. Through the use of weighted backpropagation, we push the information learned by the projection model back through the neural network to generate explanations relative to the underlying data features. In doing so, we take a step toward solving the "two black boxes" problem, as defined by Wenskovitch and North [44]. The "two black boxes" problem identifies both the deep-learning algorithm and the human cognitive process as black boxes that impede the learning process. In our method, semantic interactions with the projection allow people to express some of their cognitive processes to the machine. In return, the model presents explanations that illustrate how it uses the provided information. This creates a synergy between the machine and the human and facilitates a more complete analysis experience. This framework can be generally applied to analytics methods using deep-learning representations of data.

Feature representation choice In our method, we use ResNet18 to extract image features. However, alternative methods for feature extraction could be used. Bian et al. explored additional methods for feature extraction, including color histogram and Scale-Invariant Feature Transform [45]. We explored these methods as well but found that feature representations from convolutional neural networks provide the most meaningful projections and explanations. Additionally, while we chose to use ResNet 18, our method supports swapping in other neural network feature extractors, including those designed for specific tasks and datasets. This allows people to further customize projections of their data for the given analysis task. Additionally, our method can facilitate the comparison of different feature representations to identify the one most appropriate for a given task.

Furthermore, we note that the scope of our research contributions is focused on demonstrating the effectiveness of our interactive DR method for images, rather than comparing or evaluating our method against different image models. ResNet was chosen to demonstrate the capabilities of our interactive DR strategy and to serve as a concrete example of the application of our interactive DR model. While it is possible that other image models could influence the results, our emphasis lies in the novel aspects of our interactive DR approach, which enables users to explore and manipulate the 2D projection space, obtain visual explanations, and investigate complex visual relationships in image data. By integrating both human-mental models and pre-trained deep neural models, our method provides a novel perspective for understanding and interacting with image collections.

Other methods for explanation Our method uses weighted backpropagation to create explanations of the effects of semantic interactions. However, this method is only one can-

didate for creating explanations of interactions. There exist other methods for generating feature explanations that we can adapt to our method. For example, we also adapted Grad-CAM to consider the weights from the projection model to generate explanations [31]. We found that Grad-CAM excels when images contain multiple entities; however, it falls flat when searching for specific image features. As our method benefits from finer-grained explanations, Grad-CAM was not a suitable method. Adapting other methods for creating model explanations remains to be explored in future work.

Retaining human feedback While our method helps people explore many organizations of images and incrementally build a mental model of the underlying data, it has limited knowledge retention for iteratively fine-tuning a single model. To overcome this, we need to explore methods for incorporating learned information back into the feature extractor to update the representation to retain human feedback throughout the image sorting process, similar to Bian et al.'s method for textual data [27]. The drawback to this is that it trades fine-tuning of a single model for the ability to easily change the basis of the organization. If the user specifies contradicting information over the course of several iterations of interaction, it may confuse the model and produce a less organized layout of the images. This method and its limits remain to be explored in future work.

Scalability of explanations One outstanding challenge in designing explanations for collections of images is scalability. Methods such as ours require people to inspect individual images, to understand the important features. In the presence of large datasets, this becomes cumbersome and impractical. Currently, the only solution offered by our method is to plot the explanations themselves rather than the images, enabling people to consume them more quickly. This, however, still has drawbacks for larger datasets due to the occlusion of plotted images and the time required to visually scan through many explanations. The natural solution would be to create summary explanations for sets of images. However, images present a unique challenge in that it is non-trivial to summarize a set of images. An additional solution could be to design metrics that suggest important explanations, e.g., explanations that change substantially after an interaction, to help reduce the amount of explanations to inspect. Future work is needed to explore how to support scalable explanations of images.

8 Conclusion

In this paper, we presented an interactive dimension reduction method for exploring image data using deep-learning representations of images. Our method provides semantic interactions that allow people to incorporate their prior knowledge into the projection model. It uses custom-defined relationships to learn new projection weights optimal for respecting these relationships. Additionally, our method provides visual explanations of the effects of semantic interactions on the placement of images in the projection. These explanations illustrate the image features most important for projecting the images and illustrate the effects of interactions. We provide a real-world usage scenario and quantitative analysis to demonstrate the method's effectiveness at organizing data from human-defined similarities. Overall, we found that our method was able to capture human feedback and incorporate it into the model. Our visual explanations help bridge the gap between the feature space and the original images to illustrate the knowledge learned by the model, creating a synergy between humans and machines that facilitates a more complete analysis experience.

Acknowledgements This material is based upon work supported by the National Science Foundation under Grant # 2127309 to the Computing Research Association for the CIFellows 2021 Project. This project was funded, in part, with an integrated internal competitive grant from the College of Agriculture and Life Sciences at Virginia Tech.

References

1. Cunningham, P.: Dimension reduction. In: Machine Learning Techniques for Multimedia, pp 91–112. Springer (2008)
2. Endert, A., Chang, R., North, C., Zhou, M.: Semantic interaction: coupling cognition and computation through usable interactive analytics. IEEE Comput. Gr. Appl. **35**(4), 94–99 (2015)
3. Self, J.Z., Dowling, M., Wenskovitch, J., Crandell, I., Wang, M., House, L., et al.: Observation-level and parametric interaction for high-dimensional data analysis. ACM Trans. Interact. Intell. Syst. **8**(2), 1–36 (2018)
4. Cheng, T.Y., Huertas-Company, M., Conselice, C.J., Aragon-Salamanca, A., Robertson, B.E., Ramachandra, N.: Beyond the Hubble sequence-exploring galaxy morphology with unsupervised machine learning. Mon. Not. R. Astron. Soc. **503**(3), 4446–4465 (2021)
5. Bojarski, M., Choromanska, A., Choromanski, K., Firner, B., Jackel, L., Muller, U., et al.: Visualbackprop: efficient visualization of CNNs. Preprint at arXiv:1611.05418 (2016)
6. Han, H., Faust, R., Norambuena, B.F.K., Prabhu, R., Smith, T., Li, S., et al.: Explainable interactive projections for image data. In: Advances in Visual Computing: 17th International Symposium, ISVC 2022, San Diego, CA, USA, October 3–5, 2022, Proceedings, Part I., pp. 77–90. Springer (2022)
7. Tukey, J.W., Wilk, M.B.: Data analysis and statistics: an expository overview. In: Proceedings of the November 7-10, 1966, Fall Joint Computer Conference, pp. 695–709 (1966)
8. Cavallo, M., Demiralp, Ç.: A visual interaction framework for dimensionality reduction based data exploration. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, pp. 1–13 (2018)
9. Jeong, D.H., Ziemkiewicz, C., Fisher, B., Ribarsky, W., Chang, R.: iPCA: an interactive system for PCA-based visual analytics. In: Computer Graphics Forum, vol. 28, pp. 767–774. Wiley Online Library (2009)
10. Espadoto, M., Appleby, G., Suh, A., Cashman, D., Li, M., Scheidegger, C.E., et al.: UnProjection: leveraging inverse-projections

for visual analytics of high-dimensional data. IEEE Transactions on Visualization and Computer Graphics (2021)
11. dos Santos Amorim, E.P., Brazil, E.V., Daniels, J., Joia, P., Nonato, L.G., Sousa, M.C.: iLAMP: exploring high-dimensional spacing through backward multidimensional projection. In: 2012 IEEE Conference on Visual Analytics Science and Technology, pp. 53–62. IEEE (2012)
12. Eler, D.M., Nakazaki, M.Y., Paulovich, F.V., Santos, D.P., Andery, G.F., Oliveira, M.C.F., et al.: Visual analysis of image collections. Vis. Comput. **25**(10), 923–937 (2009)
13. Paulovich, F.V., Eler, D.M., Poco, J., Botha, C.P., Minghim, R., Nonato, L.G.: Piece wise Laplacian-based projection for interactive data exploration and organization. In: Computer Graphics Forum, vol. 30, pp. 1091–1100, Wiley Online Library (2011)
14. Joia, P., Coimbra, D., Cuminato, J.A., Paulovich, F.V., Nonato, L.G.: Local affine multidimensional projection. IEEE Trans. Vis. Comput. Gr. **17**(12), 2563–2571 (2011)
15. Mamani, G.M., Fatore, F.M., Nonato, L.G., Paulovich, F.V.: User-driven feature space transformation. In: Computer Graphics Forum, vol. 32. pp. 291–299. Wiley Online Library (2013)
16. Brown, E.T., Liu, J., Brodley, C.E., Chang, R.: Dis-function: learning distance functions interactively. In: IEEE Conference on Visual Analytics Science and Technology. vol. 2012, pp. 83–92. IEEE (2012)
17. Fujiwara, T., Wei, X., Zhao, J., Ma, K.L.: Interactive dimensionality reduction for comparative analysis. IEEE Trans. Vis. Comput. Gr. **28**(1), 758–768 (2022). https://doi.org/10.1109/TVCG.2021.3114807
18. Endert, A., Fiaux, P., North, C.: Semantic interaction for visual text analytics. In: Proc. of the SIGCHI Conference on Human Factors in Computing Systems. CHI '12. pp. 473-482. ACM, New York. Available from: https://doi.org/10.1145/2207676.2207741 (2012)
19. Endert, A., Fiaux, P., North, C.: Semantic interaction for sensemaking: inferring analytical reasoning for model steering. IEEE Trans. Vis. Comput. Gr. **18**(12), 2879–2888 (2012)
20. Dowling, M., Wycoff, N., Mayer, B., Wenskovitch, J., House, L., Polys, N., et al.: Interactive visual analytics for sensemaking with big text. Big Data Res. **16**, 49–58 (2019)
21. Dowling, M., Wenskovitch, J., Hauck, P., Binford, A., Polys, N., North, C.: A bidirectional pipeline for semantic interaction. In: Proc. Workshop on Machine Learning from User Interaction for Visualization and Analytics (at IEEE VIS 2018), vol. 11, p. 74 (2018)
22. Wang, M., Wenskovitch, J., House, L., Polys, N., North, C.: Bridging cognitive gaps between user and model in interactive dimension reduction. Vis. Inform. **5**(2), 13–25 (2021)
23. Endert, A., Han, C., Maiti, D., House, L., North, C.: Observation-level interaction with statistical models for visual analytics. In: IEEE Conference on Visual Analytics Science and Technology, vol. 2011, pp. 121–130. IEEE (2011)
24. House, L., Leman, S., Han, C.: Bayesian visual analytics: Bava. Stat. Anal. Data Min. ASA Data Sci. J. **8**(1), 1–13 (2015)
25. Leman, S.C., House, L., Maiti, D., Endert, A., North, C.: Visual to parametric interaction (v2pi). PloS One **8**(3), e50474 (2013)
26. Krokos, E., Cheng, H.C., Chang, J., Nebesh, B., Paul, C.L., Whitley, K., et al.: Enhancing deep learning with visual interactions. ACM Trans. Interact. Intell. Syst. (TiiS) **9**(1), 1–27 (2019)
27. Bian, Y., North, C.: Deepsi: interactive deep learning for semantic interaction. In: 26th International Conference on Intelligent User Interfaces, pp. 197–207 (2021)
28. Bian, Y., North, C., Krokos, E., Joseph, S.: Semantic, explanation of interactive dimensionality reduction. In: IEEE Visualization Conference (VIS), vol. 2021, pp. 26–30. IEEE (2021)
29. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: European Conference on Computer Vision, pp. 818–833. Springer (2014)
30. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2921–2929 (2016)
31. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 618–626 (2017)
32. Pirolli, P., Card, S.: The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In: Proceedings of International Conference on Intelligence Analysis, vol. 5, pp. 2–4. McLean, VA (2005)
33. Salau, A.O., Jain, S.: Feature extraction: a survey of the types, techniques, applications. In: 2019 International Conference on Signal Processing and Communication (ICSC), pp. 158–164. IEEE (2019)
34. Chen, J., Zou, L.H., Zhang, J., Dou, L.H.: The comparison and application of corner detection algorithms. J. Multimed. 4(6) (2009)
35. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proceedings of the Seventh IEEE International Conference on Computer Vision. vol. 2, pp. 1150–1157. IEEE (1999)
36. Ghosh, S.K., Biswas, B., Ghosh, A.: A novel noise removal technique influenced by deep convolutional autoencoders on mammograms. In: Deep Learning in Data Analytics. pp. 25–43. Springer (2022)
37. Yu, S., Jia, S., Xu, C.: Convolutional neural networks for hyperspectral image classification. Neurocomputing **219**, 88–98 (2017)
38. Villaret, M., et al.: Affective state-based framework for e-Learning systems. In: Artificial Intelligence Research and Development: Proceedings of the 23rd International Conference of the Catalan Association for Artificial Intelligence. vol. 339, p. 357. IOS Press (2021)
39. Baumgartl, H., Buettner, R.: Developing efficient transfer learning strategies for robust scene recognition in mobile robotics using pre-trained convolutional neural networks. Preprint at arXiv:2107.11187 (2021)
40. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
41. Han, H., Prabhu, R., Smith, T., Dhakal, K., Wei, X., Li, S., et al.: Interactive deep learning for exploratory sorting of plant images by visual phenotypes (2022)
42. Banerjee, S.: Animal image dataset. https://www.kaggle.com/datasets/iamsouravbanerjee/animal-image-dataset-90-different-animals?select=animals
43. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math. **20**, 53–65 (1987)
44. Wenskovitch, J., North, C.: Interactive AI: Designing for the 'Two Black Boxes' Problem, pp. 1–10. IEEE Computer Society, Hybrid Human-Artificial Intelligence Special Issue, Washington (2020)
45. Bian, Y., Wenskovitch, J., North, C.: Deepva: bridging cognition and computation through semantic interaction and deep learning. Preprint at arXiv:2007.15800 (2020)

Huimin Han is a researcher focusing on interactive machine learning, explainable AI, and visual analytics. She earned her master's degree from Virginia Tech advised by Dr. Chris North. During her time there, her interest in explainable AI and visual analytics took off. She focused on finding ways for people to easily work with complex data, making it easier to understand advanced machine learning. One accomplishment during this phase was Huimin's work in improving how people interact with data. She added visual explanations to help users grasp machine learning models better. By combining simple interfaces with clear explanations, she aims to make AI more understandable and approachable.

Rebecca Faust is a Postdoc in Computer Science at Virginia Tech under a Computing Innovations Fellowship. She received her Ph.D. from the University of Arizona in 2021. Her research centers around using visualization to help explain data analysis programs and workflows, as well as enable interaction for human–machine teaming.

Brian Felipe Keith Norambuena is an Assistant Professor at the Department of Computing and Systems Engineering. He obtained his Ph.D. in Computer Science and Applications at Virginia Tech in 2023. His research areas include visualization, artificial intelligence, and computational narratives. He obtained a degree in Civil Engineering in Computing and Informatics in 2016, a bachelor's degree in Mathematics in 2017, and a master's degree in Computer Engineering in 2017 from Universidad Católica del Norte.

Jiayue Lin received a Bachelor of Engineering degree in Computer Science from Virginia Tech in 2023. Currently, he is enrolled in the Accelerated BS/MS Program, pursuing an MSc thesis in Computer Science and Applications under the guidance of his advisor, Dr. Chris North. His research interests primarily revolve around visual analytics and explainable artificial intelligence.

Song Li is an Associate Professor in the School of Plant and Environmental Sciences at Virginia Tech. He received his Ph.D. in Genomics and Bioinformatics from Penn State University. His research focuses on the interface of agriculture and data science, with the aim to develop computational tools and machine learning methods that integrate and interpret large-scale data generated by advanced genomics and sensor technologies in agricultural research.

Chris North is a Professor of Computer Science at Virginia Tech, where he is also Associate Director of the Sanghani Center for AI and Data Analytics, and member of the Center for Human–Computer Interaction. His research investigates novel methods for human–AI interaction in data analytics and visualization.