Explainable Interactive Projections for Image Data

Huimin Han

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Computer Science and Applications

Christopher L. North, Chair

Lifu Huang

Song Li

December 13, 2022

Blacksburg, Virginia

Keywords: Interactive Dimension Reduction, Semantic Interaction, Explainable AI

Explainable Interactive Projections for Image Data

Huimin Han

(ABSTRACT)

Making sense of large collections of images is difficult. Dimension reductions (DR) assist by organizing images in a 2D space based on similarities, but 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 space 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. Two usage scenarios are provided to demonstrate that our method is 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 human and machine that facilitates a more complete analysis experience.

Explainable Interactive Projections for Image Data

Huimin Han

(GENERAL AUDIENCE ABSTRACT)

High-dimensional data is everywhere. A spreadsheet with many columns, text documents, images, ... ,etc. Exploring and visualizing high-dimensional data can be challenging. Dimension reduction (DR) techniques can help. High dimensional data can be projected into 3d or 2d space and visualized as a scatter plot.Additionally, DR tool can be interactive to help users better explore data and understand underlying algorithms. Designing such interactive DR tool is challenging for images. To address this problem, this thesis presents a tool that can visualize images to a 2D plot, data points that are considered similar are projected close to each other and vice versa. Users can manipulate images directly on this scatterplot-like visualization based on own knowledge to update the display, saliency maps are provided to reflect model's re-projection reasoning.

Dedication

To mom, dad and Kaiyi

Acknowledgments

Firstly, I would like to thank my advisor Dr. Christopher North for his continued support in my research study. He was and remains my best role model for a teacher, mentor, and researcher.

I would also like to thank my committee members who have helped me navigate this degree: Dr. Song Li and Dr. Lifu Huang.

Lastly, I would like to thank my great partners Rebecca Faust, Brian Keith and Ritvik Prabhu.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Dimension reduction (DR) methods are commonly used to organize data for sensemaking tasks [12]. Their rise in popularity is due to their ability to map a high-dimensional dataset to a low-dimensional space (typically 2D) while preserving meaningful structure and relationships from the original, high-dimensional data. In fact, they excel at creating low-dimensional layouts that position similar data in close proximity and provide a visual summary of the high-dimensional data. Additionally, researchers have developed interactive DR methods to allow people to provide feedback to the projection to steer it based on their prior knowledge. In particular, semantic interaction (SI) couples cognitive and computational processes by inferring meaning behind interactions and updating the model accordingly [20].

However, most of these methods have limited support for image data, treating it the same as tabular data. Images present an additional obstacle for interactive DR methods due to their complexity and lack of interpretable dimensions. Most DR methods for image sensemaking represent images as an array where each dimension represents a pixel. This representation imposes limitations on the DRs ability to identify similarities between images and inhibits the semantic interactions' ability to infer meaning from interactions. For example, Self et al.'s Andromeda uses Weighted Multidimensional Scaling (WMDS) to create an interactive DR that supports semantic interaction for model steering [42]. After an interaction, the model learns new weights on the input dimensions that infer meaning from the interaction. When a dataset has interpretable dimensions, this makes sense as a subset of the dimensions

likely corresponds to the intended concepts of the analyst, and increasing the weight of a dimension has a clear meaning in terms of the data. However, when the dimensions are pixels, this is no longer true—increasing the weight on a single pixel has no inherent meaning. To accommodate image data, an alternative representation that better captures human feedback is needed.

It is known from past research that deep neural networks are adept at extracting meaningful features from images embedding them into a new representation[11, 31]. These embeddings are commonly used for image classification tasks [1]. Classifiers built from these embeddings are quite accurate which indicates that the embeddings must be well suited for finding similarities between images. Here comes to the first research question:

- **RQ1:** How can people use these feature embeddings to create more meaningful projections of image data and capture human feedback?

In this work, an interactive DR method is presented, built from Self et al.'s Andromeda, that supports semantic interaction for sorting and organizing image data. Method in this thesis leverages the feature embeddings extracted from a convolutional neural network to project image data to a low-dimensional space using Weighted Multidimensional Scaling (WMDS). Each dimension in the embedding represents some feature of the images. method in this thesis allows people to directly manipulate the 2D representation to define similarities within the data. Using these definitions, the method learns new weights on the embedding features that best respect the defined similarities and updates the projection accordingly. Unlike pixel dimensions, the embedding features have an inherent meaning. Increasing the weight on one of the features subsequently increases the importance of that feature in the projection.

However, a significant problem still exists: while the embedding features have an inherent

meaning, that meaning is only interpretable by the machine. Thus, while updating the weights now has inherent meaning, people have no real understanding of this meaning. That brings to the second research question:

- **RQ2:** How to translate the learned weights back to the image space?

Most interactive DR methods project data from vectors of pixels. However, DR's of pixel representations fail to account for features such as lighting, distortions, and misalignment. In contrast, neural networks excel at extracting meaningful features in images and finding similarities between them [11, 31]. Thus, in this work, a neural network is used to extract features to represent the images. By using the neural feature representations, the DR methods create more meaningful organizations of image data and do not suffer from the same limitations of the pixel representation.

In addition to providing an interactive DR, the proposed approach provides explanations of semantic interactions through the use of a weighted backpropagation algorithm. A traditional visual backpropagation method is adapted for generating saliency maps [7] to apply the feature weights from the projection. Doing so creates saliency maps that emphasize the image features most influential to the projection's placement of the image. Thus, people will be able to push the information learned from the human interaction back through the network to the image space, where people can interpret it.

Method proposed in this thesis helps people organize their image data through semantic interactions and explain the effects of these interactions on the placement of images through saliency maps.

Chapter 2

Review of Literature

This 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 [44]. 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 [9, 29]. Other works explore the idea of backwards (or inverse) projections that allow people to select locations in the 2D space and generate corresponding high-dimensional representations [13, 21]. PEx-Image specifically targets image data, providing interactions for exploratory tasks, such as zooming into specific projection regions, displacing points to resolve overlapping and

displaying nearest neighbors of selected images [16].

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 [30, 35, 37]. Others learn new distance functions for MDS to update the projection to best respect user manipulations [8, 42]. 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 [22].

This 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 [42]. 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 [15].

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 bidirectional pipeline [14] 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) [46]. There are several models that can be used to solve the bi-directional transforms required to implement semantic interactions, such as Observation-Level Interaction [17], Bayesian Visual Analytics [28], and Visual to Parametric Interaction [32].

Previous work has also shown how to integrate deep learning models with semantic interaction techniques. Bian and North [4] developed a semantic interaction model for text analytics integrating traditional dimensionality reduction techniques with a BERT neural network as its core component. Bian et al. [6] 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. [7] proposed a visualization method that shows which pixels of an input image contribute the most towards 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 [49] developed a novel visualization technique that provides insight into the intermediate feature layers of a CNN in a classification task. Zhou et al. [50] 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 expla-

nation 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. [43] 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, this work seeks to fill this gap and combine interactive DR for images with explainable deep learning techniques. In particular, this work is based on the method of Bojarski et al. [7], 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.

Chapter 3

Tasks

Before discussing the detail, we first must discuss the sensemaking tasks of someone using method in this thesis. Pirolli and Card described the sensemaking process as having two primary loops: the foraging loop and the sensemaking loop [38]. 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.

Figure 3.1: An overview of the views presented by method proposed in this thesis. (a) shows the initial projection of the images. Users can select points (indicated by the green circle) and drag them to new positions in the plot. (b) shows the plot after user interactions. (c) shows the visual explanations provided by method in this thesis. Brighter regions indicate features of importance used by the projection to place the image.

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.

The proposed 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:

- Define custom similarities based on prior knowledge
- Link human and machine defined similarities
- Refine the model for downstream use

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, to create projections that better represent the underlying data. In the end, an analysis pipeline is created 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 with respect to their analysis task, such as what categories of images they expect to exist within the data. For example, consider a projection of edamame plants. When analyzing the data, one task may be to sort the images of diseased pods from the healthy ones. Static dimension reduction plots, may or may not adequately reflect this prior knowledge. In the previous example, the person analyzing the data knows what defines healthy vs diseased, but the model may not naturally recognize these differences. To enable hypothesis testing for the analysis task, people must be able to steer the projection to define similarities in the data in a way that reflects their prior knowledge. With method in this thesis, 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. In the edamame example, the analyst drags some of the diseased pods to one corner and some healthy pods to another. This tells the projection model that images like these should be placed in separate groups. Through repeated iterations of these steering interactions, people incrementally reshape the model to present the data in a way that better reflects their prior knowledge and enables their analysis tasks.

3.2 Link human and machine defined similarities

The previous tasks focus on teaching the projection model to incorporate human knowledge into it. However, when the model updates, it may or may not use that knowledge in precisely the intended way. The machine may identify the intended features or it may find

other features that re-organize the projection while respecting the defined similarities. For example, consider organizing the edamame pods by the number of seeds in the pod, 1, 2, or 3. The analyst performs several interactions to convey this information. However, perhaps all of the pods with three seeds that they moved also were diseased. The projection may reorganize in a meaningful way, but it may also use unintended features, such as diseased spots, to organize the pods in a way that does not quite match the task. Thus, explanations of the image features used to place images may illustrate unexpected relationships between images. In this thesis, we 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.

3.3 Refine the model for downstream use

Repeated iterations of the previous tasks create a refined schema to represent the underlying data. However, the sensemaking process does not stop there. After refining the schema to suit the analysis task, the projection weights can be exported and fed into downstream models for further tasks, such as classification. Rather than ending the analysis pipeline with method in this thesis, it can serve as an intermediate step that helps organize and refine the schema and model weights for exportation into a downstream model or task. For example, after steering the projection to organize the pods based on the number of seeds, the analyst may want to export the model parameters to feed into a classifier to help it better identify pods of each type.

Chapter 4

Methodology and Workflows

In this section, the details of the main components of method in this thesis are described: feature extraction, interactive DR, and explainability visualizations.

4.1 Feature Extraction

Feature extraction is an important technique in computer vision widely used for tasks such as object detection and image classification [41]. Existing feature extraction methods for image data include traditional approaches such as Harris Corner Detection [10] and Scale-Invariant Feature Transform (SIFT) [34]. Recently, deep learning models have become popular for feature extraction in images [23]. In particular, Convolutional Neural Networks (CNN) have shown great power in image-related tasks [48]. Thus, using CNNs has become the standard in feature extraction [45].

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 [3]. For this research, we use pre-trained ResNet18 [26] as a fixed feature extractor to generate features 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:

$$\mathcal{X} = ResNet_{\mathsf{pre-trained}}(\mathcal{D}) \tag{4.1}$$

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.

4.2 Interactive Dimension Reduction

To facilitate interactive dimension reduction, we use Weighted Multidimensional Scaling (WMDS) for the forward projection and inverse WMDS (WMDS^{-1}) to update the projection after semantic interactions, as originally described in Andromeda [42].

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

$$y = \underset{y_1,\ldots y_n}{\arg\min} \sqrt{\sum_{i<j\leq N} (dist_L(y_i, y_j) - dist_H(w, x_i, x_j))^2} \tag{4.2}$$

where N is the number of points in the dataset, $dist_L(y_i, y_j)$ is the low-dimensional distance between y_i and y_j and $dist_H(w, x_i, x_j)$ is the weighted high dimensional distance between the feature representations x_i and x_j, given the dimension weights w.

For the initial projection, we initialize w with equal weights for every dimension, relying solely on the raw image features to organize the images. After a user 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:

$$w = \underset{w_1,\ldots w_d}{\arg\min} \sqrt{\frac{(\sum_{i<j\leq N}(dist_L(y_i^*, y_j^*) - dist_H(w, x_i, x_j))^2}{\sum_{i<j\leq N} dist_H(w, x_i, x_j)^2}} \tag{4.3}$$

This equation produces a vector of dimension weights that best respects the 2D similarities specified through the interactions. Additionally, we normalize the weight vector to sum to 1, so as to normalize the HD distances to a roughly constant size space. We then re-project the images using equation 4.2 with the updated weights to create a layout that incorporates human feedback.

4.3 Explainability Visualization

One of the central problems of using deep learning feature representations is the lack of context from the original dataset. To overcome this problem, we propose a weighted visual backpropagation method to provide visual feedback and explanations of the information learned by the projection. This feedback enables people to better complete their tasks as they iterate and refine their sensemaking schemas.

The proposed weighted visual backpropagation method is based on the original visual backpropagation method proposed by Bojarski et al.[7]. This approach did not focus on computing gradients, but instead, computes the actual contribution of neurons, making the backpropagation process fast and efficient. The extension to this method is highlighted in red in Figure 4.1.

To implement the extended 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 dimension reduction loop to back-propagate 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 Figure 4.1, is deconvolved with the same filter size and stride as the convolutional layer immediately preceding it. This scales the feature map up to match the size of the feature map in the previous layer. Then the deconvolved feature map is point-wise multiplied by the averaged single feature map of the previous layer. This process is repeated until we reach the input image. Figure 4.1 shows the entire backpropagation process.

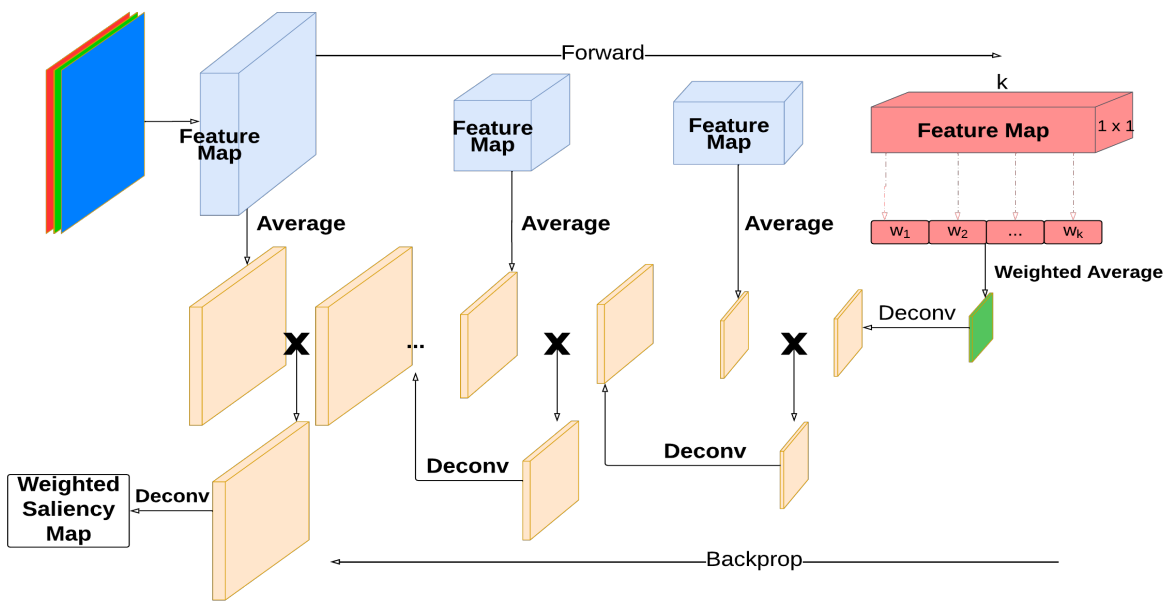


Figure 4.1: Weighted Visual Backpropagation Process

The notation is kept consistent with Bojarski et al.[7]. Note, only the modification to their method will be described. For full details on visual backpropagation, 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 neuron 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:

$$\theta^{\mathcal{N}}_{VBP}(i) = c * \gamma(i) \sum_{P \in \mathcal{P}} \prod_{e \in P} a_e \tag{4.4}$$

For method in this thesis, 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 et as the edge that connects nodes from the convolutional layer $(t-1)$ to the convolutional layer t. Let k denote the kernels for each layer. Then, the contribution of the input pixel i calculated by the Weighted Visual Backpropagation method is defined as

$$\theta^{\mathcal{N}}_{WVBP}(i) = c * \gamma(i) \sum_{P \in \mathcal{P}} \prod_{e \in P} a_{et} \tag{4.5}$$

where

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

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

Chapter 5

Usage Scenarios

5.1 Animals

In this scenario, we use a dataset of images of animals from Kaggle[2]. This dataset consists of 5400 animal images in 90 different classes. For the task, we sampled a small subset of this data, using only 5 classes of animals—horse, goose, shark, snake, and eagle—with 5 or 6 images per class. Figure 5.1 illustrates this usage scenario.

5.1.1 Image Sorting Task

In addition to the animal, several of the images also contain a human. A natural task for this data is then to define images containing humans as similar to one another, and dissimilar from images containing only animals.

5.1.2 Analysis Scenario

After loading the data, method in this thesis creates the initial projection of the images, shown in Figure 5.1(a). 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

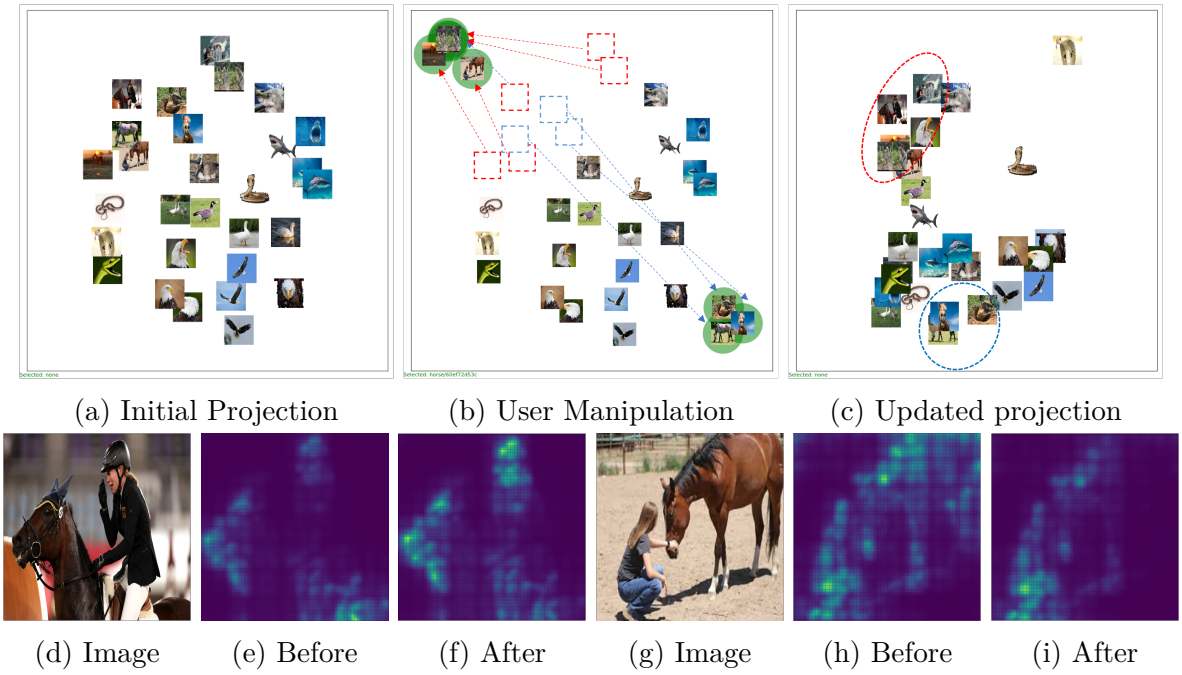


Figure 5.1: Usage scenario on the animals dataset: (a,b,c) show the process for exploratory analysis on a small subset of images. In (b) the user drags the "human and horse" images apart from "horse" images to emphasize the "human" object. In the updated projection (c) the animals are projected near the bottom and images containing "humans" are clustered at the top (circled in red). (e,h) show the saliency maps before the interactions, and (f,i) show the maps after the interactions, now with greater attention on the "human" object.

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 so, we drag the "human and horse" images apart from "single horse" images as shown in Figure 5.1(b). After this, the underlying model learns the current user-defined layout and updates the entire projection based on the learned weights. Figure 5.1(c) 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.

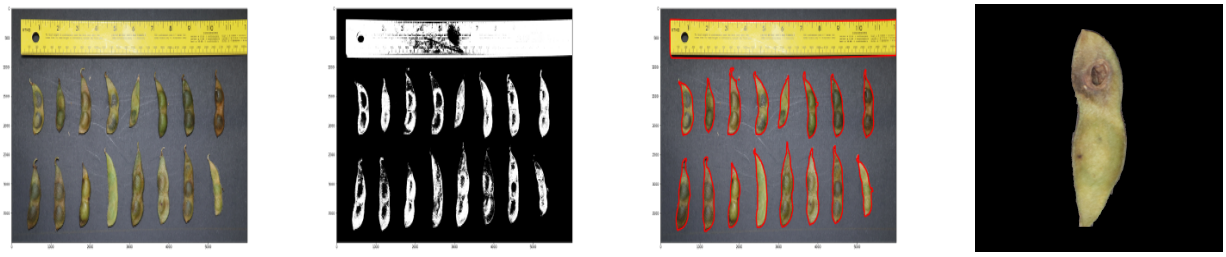
5.1.3 Verifying the Learned Information

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 Figure 5.1 (d) - (i). To illustrate this, we select two of the images containing humans and horses, shown in Figure 5.1 (d,g). Before the user manipulates images, the underlying model projected images mainly based on animal content in the images as shown in Figure 5.1(e,h). 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 and less attention on the horses (or animals in general) as shown in Figure 5.1(f,i). Using the visual explanations, we clearly see that the projection adequately inferred the meaning behind the interactions.

5.2 Edamame Pods

5.2.1 Dataset and Preprocessing

Images used in this paper were collected by the Li Lab of Applied Machine Learning in Genomics and Phenomics at Virginia Tech [33]. This dataset comprises ready-to-harvest, late-to-harvest, and diseased pod images. Figure 5.2 shows the sample raw data and image pre-processing results. We used an improved vegetation index, Excess Green minus Excess Red (ExG- ExR) [36], to identify pods for our data sets. ExR was subtracted from ExG with a zero threshold to create the ExG-ExR binary image. After computing a binary image from vegetation indices, we applied several morphological transformations [24, 39]. We used dilation to increase the object area and closing and opening, which cleaned background noise by imputing missing pixel values. Finally, after vegetation indices and morphological transformations, we obtained a binary image mask with pods as white and background as black. Pods were detected by finding the contours of these masks.

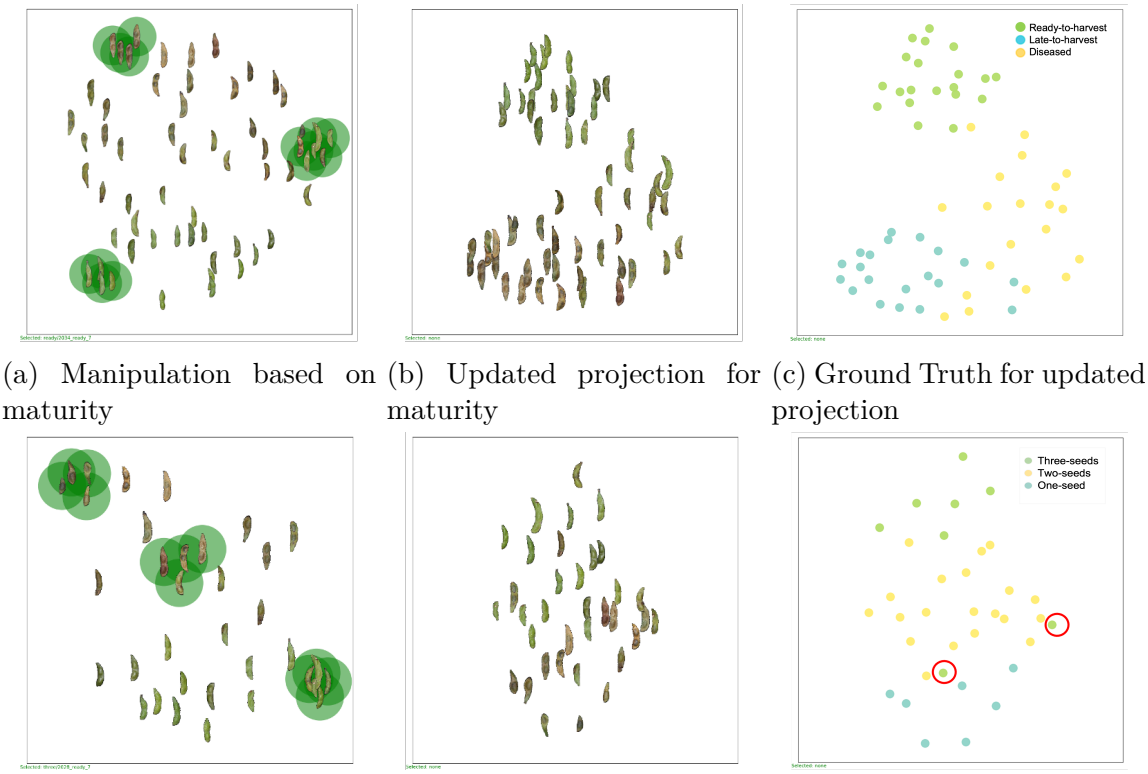


(a) Sample Raw Data (b) Binary Image (c) Object Detection (d) Segmented Image

Figure 5.2: Sample raw data and preprocessing results for a diseased pod

This usage scenario was developed with collaborators in the plant sciences department [25]. The 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. They initially trained a model to classify pods based on their maturity

stage: ready to harvest, late to harvest, and diseased. However, when sorting 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 the ones in this thesis 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 example, a subset of their edamame pod dataset containing 60 images is used, with 20 images per maturity stage.



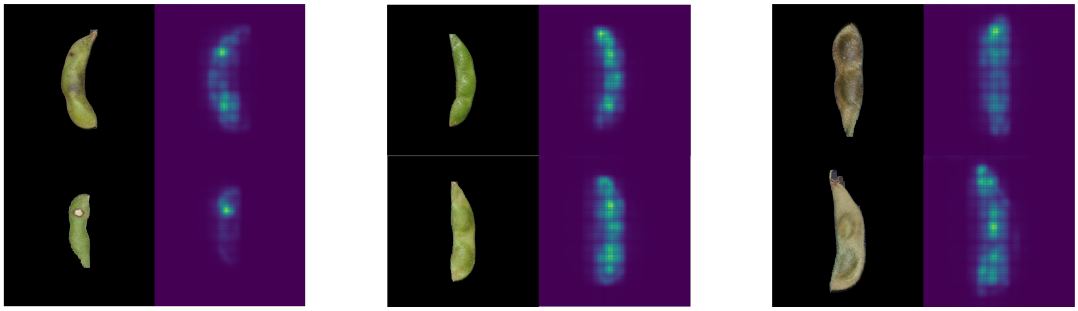
(a) Manipulation based on maturity (b) Updated projection for maturity (c) Ground Truth for updated projection

(d) Manipulation based on number of seeds (e) Updated projection for number of seeds (f) Ground truth for updated projection

Figure 5.3: Usage scenario on the edamame pods dataset: (a,b,c) show an example of an interactive task based on the maturity stage of the pods, (d,e,f) show an example of an interactive task based on the number of seeds of the pods.

5.2.2 Maturity Stage

The maturity stage of each pod is defined as either diseased, late-to-harvest, or ready-to-harvest. The maturity stage is a phenotype that can be determined by trained observers. Here, we test if method in this thesis can sort the images according to these phenotypes and whether the features captured by the model to separate the images are related to the underlying phenotypes. The edamame pods data set is displayed on the 2D projection. Then, we observe the visual phenotypes for maturity and interactively drag a subset of pods (highlighted in green) in order to group them into 3 clusters according to the desired phenotype categories, shown in Figure 5.3(a). We hypothesized that, through this interaction, the underlying model would learn new weights for the feature space that satisfy the newly defined projection and properly capture the user's mental model of pod maturity.



(a) "Diseased" (b) "Ready-to-harvest" (c) "Late-to-harvest"

Figure 5.4: Explanations of important features for "Diseased","Ready-to-harvest" and 'Late-to-harvest" pods.

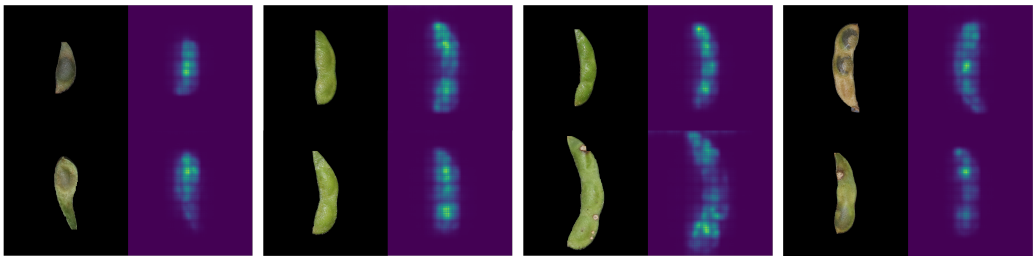
Figure 5.3(b) shows the updated projection, which produced three main clusters of pods according to their maturity stage. The ground truth of the images is shown in Figure 5.3(c). This indicates that the desired phenotypes of each pod were effectively captured by the weighted features. Thus, the model successfully learned a model of edamame pod maturity.

The explainable feature visualizations of specific pods depict the most important visual features in the re-grouping as learned by the interactive model. In Figure 5.4(a) we see that

one of the more important visual features learned by the model to determine the disease phenotype is a salient discolored spot. Similarly, in Figure 5.4(b,c), the model focuses on image areas correlated to important features of each pod. This provides us with insight into which parts of the pod are important for visually discerning a diseased, late-to-harvest, or ready-to-harvest product. Furthermore, these results also provide a comparison between human perception and machine learning.

5.2.3 Number of Pods

For the same pods dataset, we also want to explore different visual phenotypes. In particular, the number of seeds per pod is an important phenotype that potentially affects the consumers' acceptance of the product. 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 3 clusters according to the number of seeds (1,2 or 3), as shown in Figure 5.3(d). We hypothesize that by dragging a subset of the image data, the underlying model will learn the weights for the feature spaces that satisfy the user-defined projection based on the number of seeds.



(a) "One-seed" (b) "Two-seeds" (c) "Three-seeds" (d) Misprojected

Figure 5.5: Explanations of important features for the "number-of-seeds" task.

Figure 5.3(e) shows the updated projection. We find that the "number of seeds" phenotype is captured well by the weighted features learned by Andromeda. Figure 5.3(f) 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-seeds" pods. To learn more about why these two pods are mis-projected, we explore the visual feature explanations for each group. Figure 5.5(a,b,c) 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 mis-projected pods, they are either dominated by the disease spot or do not have the obvious shape of three seeded pods, as shown by Figure 5.5(d).

Chapter 6

Quantitative Analysis

In addition to the use cases, we provide a quantitative analysis to assess method in this thesis's ability to organize the images based on human feedback and evaluate the number of interactions necessary to produce a desirable organization. Ultimately, the goal of using this method is to steer the projection to create an image organization that reflects their prior knowledge. As such, a natural way to evaluate it is to measure how well the updated projection separates images into clusters after user interactions organize images by class. Additionally, we evaluate how many interactions per class are necessary to reach a well-clustered layout. The remainder of this section describes the details of the evaluation.

6.1 Method

The experimental design stems from the simulation experiments in [4]. To evaluate method in this thesis, 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 a dataset of images of animals[2]. The dataset contains 300 images with 5 classes of images (horse, goose, eagle, shark, snake), giving 60 images per category. Using this dataset, the simulated analyst simply wants to organize the images such that the animals in the same class are placed near each other.

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 (Figure 6.1(a)), (3) learn new weights using WMDS^{-}1 that respect the simulated interactions and project the whole dataset using those weights (Figure 6.1(b) and (c)), and last, (4) use the layout evaluator to measure the performance of the resulting layout. Steps (1) and (3) are described above in Chapter 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).

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:

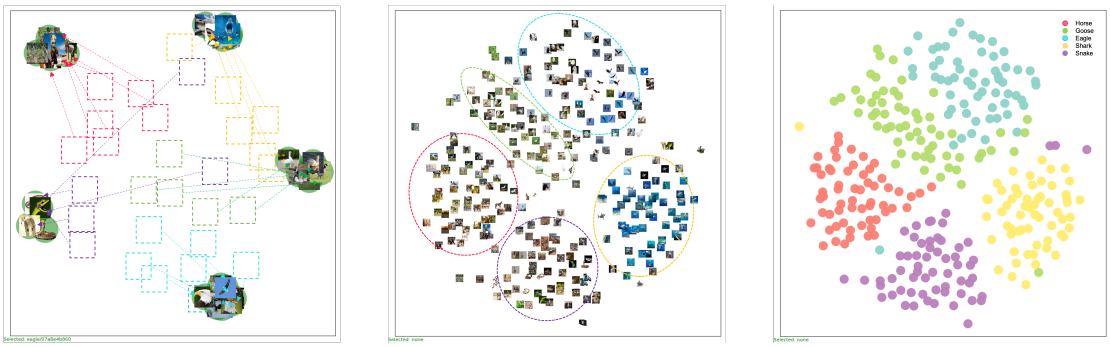
$$||x_i - x_j|| = \begin{cases} 0 & \text{if } x_i \text{ and } x_j \text{ are from same class} \\ 100 * \sqrt{2} & \text{otherwise} \end{cases}$$

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 ($100\sqrt{2}$) to teach the model that those images are dissimilar from one another. Figure 6.1 (a) provides an example of the semantic interaction that the interaction simulator is mimicking.

Layout Evaluator

After simulating the interactions, the simulation engine uses method in this thesis to learn new weights that account for the relationships defined by the interactions and projects the entire dataset using these weights. To evaluate the new layout, we calculate the Silhouette score of the clustering [40]. The Silhouette score evaluates a clustering on 2 bases: tightness and separation. It returns a value from -1 to 1, where values near zero indicate overlapping clusters, negative values indicate mis-assigned data, and a positive score indicates how tight and well separated the clusters are.

In the setting, the tightness component is less important, as tight clusters may actually be more of a hindrance than a benefit. In a DR plot, the spread of clusters may contain meaningful information that may be overlooked if the cluster is too tight. As a result, while the updated DR plots show distinct clusters of data, their Silhouette scores appear mediocre due to the lack of tightness. In this setting, a Silhouette score of around .5 actually provides a well-organized layout as exemplified by the projection Figure 6.1 (b) and (c) with a Silhouette score of 0.455.



(a) Simulated analyst fully organizes a subset of images.

(b) The learned weights are applied to the full dataset.

(c) The ground truth and performance on the full dataset.

Figure 6.1: Example of the simulation process. In (a), the analyst organizes a sample of images from each relevant label and method in this thesis 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.

6.2 Results

Figure 6.2 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 method steadily increases in its ability to organize the points. While the performance continues to increase, we see that after interacting with around 5-10 points per category, the benefits of moving more points become marginal. Figures 6.1(b) and (c) show an example layout after a user moves 5 points per class. We can see that, although the Silhouette score is only 0.455, by moving relatively few points from each class to define similarities in the dataset, method in this thesis creates a layout that respects these relationships and effectively applies them to the greater dataset.

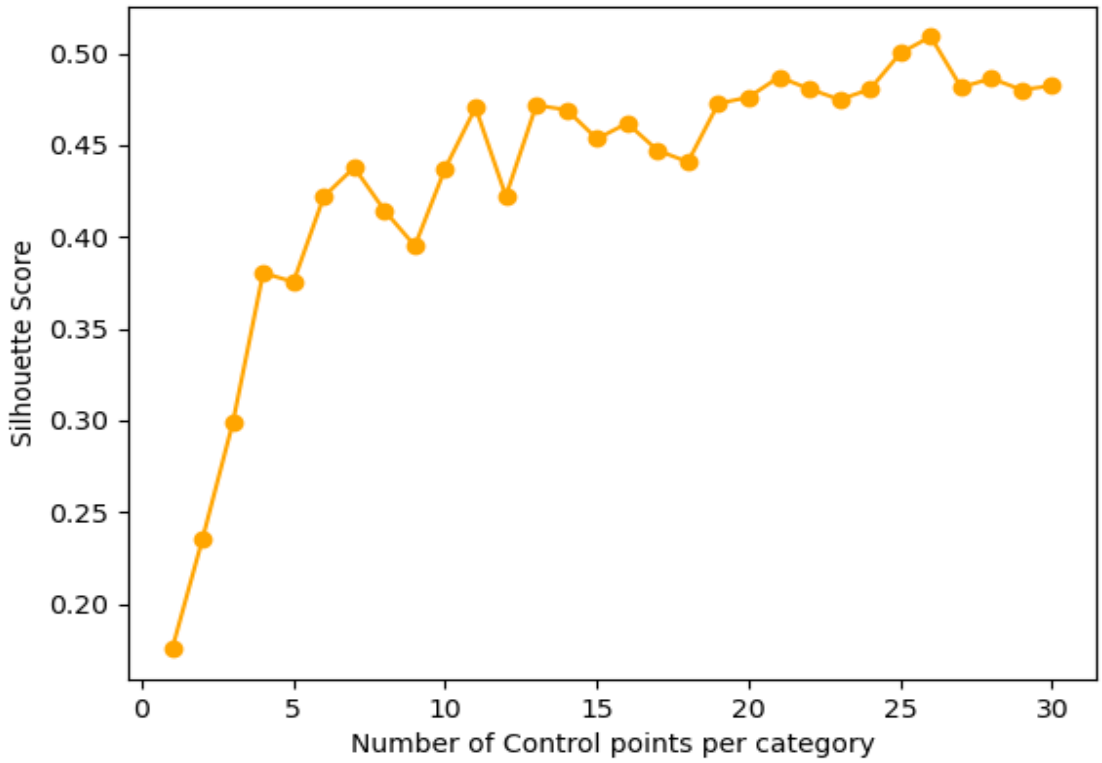


Figure 6.2: The silhouette score of the projection layout over the number of control points moved per category.

Chapter 7

Discussion

7.1 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, method in this thesis 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 towards solving the "two black boxes" problem, as defined by Wenskovitch and North [47]. 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 method in this thesis, 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.

7.2 Interactive DR as a Precursor to Classification

Method in this work projects the feature representation extracted from deep neural networks to organize and interact with image data. Historically, these image representations are used to build classification models that bin image data into pre-defined classes. While these models are useful for downstream classification tasks, they provide limited assistance for exploratory tasks as they only present a discrete, binned view of the underlying data. In contrast, interactive DR methods like the one in this thesis presents a continuous view of the data in the 2D space which provides more information about the underlying data. While the DR plot may still identify clusters of data (equivalent to the classifier bins), the 2D proximity of clusters to one another and the spread of images within a cluster may present meaningful information about similarities in the underlying data. Thus, with a continuous view of the data, people can gain greater insight into the behavior and similarities of the data. With the ability to steer the projection model, people can generate data organizations that better fit their task and funnel this information into downstream classification models (e.g. by importing the projection weights into the classification model). Thus, interactive DR methods provide finer-grained detail about the underlying data and can serve as an exploratory step toward building downstream classifiers.

7.3 Feature Representation Choice

In this work, ResNet18 is used 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 [5]. We explored these methods as well but found that feature representations from convolutional neural

networks provide the most meaningful projections and explanations. However, there exist other neural network feature extractors besides ResNet18. The design of method in this thesis easily allows people to swap in different CNN 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, method in this thesis can facilitate the comparison of different feature representations to identify the one most appropriate for a given task.

7.4 Other Methods for Explanation

The method in this thesis uses weighted backpropagation to create explanations of the effects of semantic interactions. However, this method is only one candidate for creating explanations of interactions. There exist other methods for generating feature explanations that we can adapt to method in this thesis. For example, we also adapted Grad-CAM to consider the weights from the projection model to generate explanations [43]. We found that Grad-CAM excels when images contained multiple entities, however, it falls flat when searching for specific image features. As method in this thesis 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.

7.5 Retaining Human Feedback

While method in this thesis 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 representa-

tion to retain human feedback throughout the image sorting process, similar to Bian et al.'s method for textual data [4]. 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 interaction iterations, 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.

Chapter 8

Conclusions and Future Work

8.1 Conclusions

In this work, an interactive dimension reduction method for sorting and organizing image data using deep learning representations of images is presented. Method in this thesis 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, method in this thesis provides visual explanations of the effects of semantic interactions on the projections placement of images. These explanations illustrate the image features most important for updating the projection weights after semantic interactions. To evaluate method in this thesis, two real world usage scenarios as well as a quantitative evaluation of the method's effectiveness at organizing data from human-defined similarities were provided. Overall, the work in this thesis is able to capture human feedback and incorporate it into the model. Doing so allows people to steer projections to better fit their tasks. Additionally, the 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 the human and the machine that facilitates a more complete analysis experience.

Table 8.1: Workflow Components Options

Feature Embedding Model	Dimension Reduction	Visual Explainations
Resnet-18	**MDS**	**Visual Backprop**
VGG-16	Additional projection layers	Guided Grad-cam
Other CNNs		Guided Backprop

8.2 Future Work

8.2.1 Embeddings Extracted from other CNN Models

As mentioned in 4.1, pre-trained ResNet18 is used as a fixed feature extractor, we can visualize extracted features using saliency map. Extension work from this thesis such as a comparision between features extracted from other CNN models can help understand certain architectures of CNN models. We didn't leverage very deep neural network considering performance saturation with more costs in time and computation though a deeper network may be able to extract more abstract image features matching users semantic meanings, further explorations on this trade-off would be helpful on model selection for our design.

8.2.2 Fine tune DNN model with user-defined similarities

Current design is to extract features from a pre-trained model by removing fully connection layer and freezing model parameters, using pre-trained model as a fixed feature extractor assuming image datasets' features can be well-captured by the pre-trained network. Yet this may not bbe true if the pre-trained model has no prior knowledge about the datasets at all. For future work, we can fine tune the pre-trained model efficiently with users' feedback by customized loss backward.

DNN with MDS: As shown in Table 8.1, components for the current workflow is bolded. As discussed previosuly, dimension reduction technique is used to display data in a 2D space enabling users to interact with data. Users can manipulate data directly in 2D space, inverse WMDS algorithm will generate the corresponding weighted high-dimensional space. Instead of having parameters in dimension reduction algorithms updated only, we can unfreeze DNN's parameters, the loss to backward to the DNN model (fine-tuning the model parameters) would be the sum of pairwise distances loss between the original high-dimensional space and the updated weighted high-dimensional space.

DNN with additional projection layers: Instead of using MDS algorithm to project image features into 2D space, we can also change DNN model architecture and add additional customized layers to the pre-trained model, enabling the DNN model to output 2D representations directly. Then users can manipulate the 2D data still to fine-tuning the DNN model. Distance in the output latent space need further explorations.

8.2.3 Fine tune DNN model with guided visual explainations:

Recently, scholars have proposed human-in-th-loop systems to guide DNN's attention, addressing dataset bias in deep learning. Xiaoran et al.'s [27] work proposed human-in-th-loop fine-tuning method enabling users to click on attention map then backward customized loss adding weighted guidance loss to the original model loss. In our workflow, we can further enable users to click on the saliency map increasing or decreasing weights on clicked area to guide the fine-tuning process.

Bibliography

[1] Ben Athiwaratkun and Keegan Kang. Feature representation in convolutional neural networks. *arXiv preprint arXiv:1507.02313*, 2015.

[2] Sourav Banerjee. Animal image dataset. https://www.kaggle.com/datasets/iamsouravbanerjee/animal-image-dataset-90-different-animals?select=animals.

[3] Hermann Baumgartl and Ricardo Buettner. Developing efficient transfer learning strategies for robust scene recognition in mobile robotics using pre-trained convolutional neural networks. *arXiv preprint arXiv:2107.11187*, 2021.

[4] Yali Bian and Chris North. Deepsi: Interactive deep learning for semantic interaction. In *26th International Conference on Intelligent User Interfaces*, pages 197–207, 2021.

[5] Yali Bian, John Wenskovitch, and Chris North. Deepva: Bridging cognition and computation through semantic interaction and deep learning. *arXiv preprint arXiv:2007.15800*, 2020.

[6] Yali Bian, Chris North, Eric Krokos, and Sarah Joseph. Semantic explanation of interactive dimensionality reduction. In *2021 IEEE Visualization Conference (VIS)*, pages 26–30. IEEE, 2021.

[7] Mariusz Bojarski, Anna Choromanska, Krzysztof Choromanski, Bernhard Firner, Larry Jackel, Urs Muller, and Karol Zieba. Visualbackprop: visualizing cnns for autonomous driving. *arXiv preprint arXiv:1611.05418*, 2, 2016.

[8] Eli T Brown, Jingjing Liu, Carla E Brodley, and Remco Chang. Dis-function: Learning distance functions interactively. In *2012 IEEE conference on visual analytics science and technology*, pages 83–92. IEEE, 2012.

[9] Marco Cavallo and Çağatay Demiralp. A visual interaction framework for dimensionality reduction based data exploration. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2018.

[10] Jie Chen, Li-hui Zou, Juan Zhang, and Li-hua Dou. The comparison and application of corner detection algorithms. *Journal of multimedia*, 4(6), 2009.

[11] Ting-Yun Cheng, Marc Huertas-Company, Christopher J Conselice, Alfonso Aragon-Salamanca, Brant E Robertson, and Nesar Ramachandra. Beyond the hubble sequence–exploring galaxy morphology with unsupervised machine learning. *Monthly Notices of the Royal Astronomical Society*, 503(3):4446–4465, 2021.

[12] Pádraig Cunningham. Dimension reduction. In *Machine learning techniques for multimedia*, pages 91–112. Springer, 2008.

[13] Elisa Portes dos Santos Amorim, Emilio Vital Brazil, Joel Daniels, Paulo Joia, Luis Gustavo Nonato, and Mario Costa Sousa. ilamp: Exploring high-dimensional spacing through backward multidimensional projection. In *2012 IEEE Conference on Visual Analytics Science and Technology*, pages 53–62. IEEE, 2012.

[14] Michelle Dowling, John Wenskovitch, Peter Hauck, Adam Binford, Nicholas Polys, and Chris North. A bidirectional pipeline for semantic interaction. In *Proc. Workshop on Machine Learning from User Interaction for Visualization and Analytics (at IEEE VIS 2018)*, volume 11, page 74, 2018.

[15] Michelle Dowling, Nathan Wycoff, Brian Mayer, John Wenskovitch, Leanna House,

Nicholas Polys, Chris North, and Peter Hauck. Interactive visual analytics for sensemaking with big text. *Big Data Research*, 16:49–58, 2019.

[16] Danilo M Eler, Marcel Y Nakazaki, Fernando V Paulovich, Davi P Santos, Gabriel F Andery, Maria Cristina F Oliveira, João Batista Neto, and Rosane Minghim. Visual analysis of image collections. *The Visual Computer*, 25(10):923–937, 2009.

[17] Alex Endert, Chao Han, Dipayan Maiti, Leanna House, and Chris North. Observation-level interaction with statistical models for visual analytics. In *2011 IEEE conference on visual analytics science and technology (VAST)*, pages 121–130. IEEE, 2011.

[18] Alex Endert, Patrick Fiaux, and Chris North. Semantic interaction for visual text analytics. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, page 473–482, New York, NY, USA, 2012. ACM. ISBN 9781450310154. doi: 10.1145/2207676.2207741. URL https://doi.org/10.1145/2207676.2207741.

[19] Alex Endert, Patrick Fiaux, and Chris North. Semantic interaction for sensemaking: inferring analytical reasoning for model steering. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2879–2888, 2012.

[20] Alex Endert, Remco Chang, Chris North, and Michelle Zhou. Semantic interaction: Coupling cognition and computation through usable interactive analytics. *IEEE Computer Graphics and Applications*, 35(4):94–99, 2015.

[21] Mateus Espadoto, Gabriel Appleby, Ashley Suh, Dylan Cashman, Mingwei Li, Carlos E Scheidegger, Erik Wesley Anderson, Remco Chang, and Alexandru Cristian Telea. Unprojection: Leveraging inverse-projections for visual analytics of high-dimensional data. *IEEE Transactions on Visualization and Computer Graphics*, 2021.

[22] Takanori Fujiwara, Xinhai Wei, Jian Zhao, and Kwan-Liu Ma. Interactive dimensionality reduction for comparative analysis. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):758–768, 2022. doi: 10.1109/TVCG.2021.3114807.

[23] Swarup Kr Ghosh, Biswajit Biswas, and Anupam Ghosh. A novel noise removal technique influenced by deep convolutional autoencoders on mammograms. In *Deep Learning in Data Analytics*, pages 25–43. Springer, 2022.

[24] Joseph Gil and Ron Kimmel. Efficient dilation, erosion, opening and closing algorithms. In *ISMM*, 2000.

[25] Huimin Han, Ritvik Prabhu, Timothy Smith, Kshitiz Dhakal, Xing Wei, Song Li, and Chris North. Interactive deep learning for exploratory sorting of plantimages by visual phenotypes. 2022.

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[27] Yi He, Xi Yang, Chia-Ming Chang, Haoran Xie, and Takeo Igarashi. Efficient human-in-the-loop system for guiding dnns attention, 2022. URL https://arxiv.org/abs/2206.05981.

[28] Leanna House, Scotland Leman, and Chao Han. Bayesian visual analytics: Bava. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 8(1):1–13, 2015.

[29] Dong Hyun Jeong, Caroline Ziemkiewicz, Brian Fisher, William Ribarsky, and Remco Chang. ipca: An interactive system for pca-based visual analytics. In *Computer Graphics Forum*, volume 28, pages 767–774. Wiley Online Library, 2009.

[30] Paulo Joia, Danilo Coimbra, Jose A Cuminato, Fernando V Paulovich, and Luis G Nonato. Local affine multidimensional projection. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2563–2571, 2011.

[31] Majeed Kassis, Jumana Nassour, and Jihad El-Sana. Alignment of historical handwritten manuscripts using siamese neural network. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 293–298. IEEE, 2017.

[32] Scotland C Leman, Leanna House, Dipayan Maiti, Alex Endert, and Chris North. Visual to parametric interaction (v2pi). *PloS one*, 8(3):e50474, 2013.

[33] Song Li. Li lab of applied machine learning in genomics and phenomics. https://lilabatvt.github.io/.

[34] David G Lowe. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1150–1157. Ieee, 1999.

[35] Gladys MH Mamani, Francisco M Fatore, Luis Gustavo Nonato, and Fernando Vieira Paulovich. User-driven feature space transformation. In *Computer Graphics Forum*, volume 32, pages 291–299. Wiley Online Library, 2013.

[36] George Meyer and Joao Camargo Neto. Verification of color vegetation indices for automated crop imaging applications. *Computers and Electronics in Agriculture*, 63: 282–293, 10 2008. doi: 10.1016/j.compag.2008.03.009.

[37] Fernando Vieira Paulovich, Danilo Medeiros Eler, Jorge Poco, Charl P Botha, Rosane Minghim, and Luis Gustavo Nonato. Piece wise laplacian-based projection for interac-

tive data exploration and organization. In *Computer Graphics Forum*, volume 30, pages 1091–1100. Wiley Online Library, 2011.

[38] Peter Pirolli and Stuart Card. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proceedings of international conference on intelligence analysis*, volume 5, pages 2–4. McLean, VA, USA, 2005.

[39] A.M Raid, Wael Khedr, Mohamed El-dosuky, and Mona Aoud. Image restoration based on morphological operations. *International Journal of Computer Science, Engineering and Information Technology*, 4:9–21, 07 2014. doi: 10.5121/ijcseit.2014.4302.

[40] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[41] Ayodeji Olalekan Salau and Shruti Jain. Feature extraction: a survey of the types, techniques, applications. In *2019 International Conference on Signal Processing and Communication (ICSC)*, pages 158–164. IEEE, 2019.

[42] Jessica Zeitz Self, Michelle Dowling, John Wenskovitch, Ian Crandell, Ming Wang, Leanna House, Scotland Leman, and Chris North. Observation-level and parametric interaction for high-dimensional data analysis. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 8(2):1–36, 2018.

[43] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[44] John W Tukey and Martin B Wilk. Data analysis and statistics: an expository overview.

In *Proceedings of the November 7-10, 1966, fall joint computer conference*, pages 695–709, 1966.

[45] M Villaret 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*, volume 339, page 357. IOS Press, 2021.

[46] Ming Wang, John Wenskovitch, Leanna House, Nicholas Polys, and Chris North. Bridging cognitive gaps between user and model in interactive dimension reduction. *Visual Informatics*, 5(2):13–25, 2021.

[47] John Wenskovitch and Chris North. Interactive ai: Designing for the 'two black boxes' problem. *Hybrid Human-Artificial Intelligence Special Issue (Washington, United States: IEEE Computer Society*, pages 1–10, 2020.

[48] Shiqi Yu, Sen Jia, and Chunyan Xu. Convolutional neural networks for hyperspectral image classification. *Neurocomputing*, 219:88–98, 2017.

[49] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[50] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.