ImageSI: Semantic Interaction for Deep Learning Image Projections

Jiayue Lin

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

Rebecca J. Faust

Lifu Huang

May 1, 2024

Blacksburg, Virginia

Keywords: Semantic Interaction, Deep Learning, Dimension Reduction, Images

ImageSI: Semantic Interaction for Deep Learning Image Projections

Jiayue Lin

ABSTRACT

Interactive deep learning frameworks are crucial for effectively exploring and analyzing complex image datasets in visual analytics. However, existing approaches often face challenges related to inference accuracy and adaptability. To address these issues, we propose ImageSI, a framework integrating deep learning models with semantic interaction techniques for interactive image data analysis. Unlike traditional methods, ImageSI directly incorporates user feedback into the image model, updating underlying embeddings through customized loss functions, thereby enhancing the performance of dimension reduction tasks. We introduce three variations of ImageSI, $\text{ImageSI}_{\text{MDS}^{-1}}$, prioritizing explicit pairwise relationships from user interaction, and $\text{ImageSI}_{\text{DRTriplet}}$ and $\text{ImageSI}_{\text{PHTriplet}}$, emphasizing clustering by defining groups of images based on user input. Through usage scenarios and quantitative analyses centered on algorithms, we demonstrate the superior performance of $\text{ImageSI}_{\text{DRTriplet}}$ and $\text{ImageSI}_{\text{MDS}^{-1}}$ in terms of inference accuracy and interaction efficiency. Moreover, $\text{ImageSI}_{\text{PHTriplet}}$ shows competitive results. The baseline model, WMDS^{-1}, generally exhibits lower performance metrics.

ImageSI: Semantic Interaction for Deep Learning Image Projections

Jiayue Lin

GENERAL AUDIENCE ABSTRACT

Interactive deep learning frameworks are crucial for effectively exploring and analyzing complex image datasets in visual analytics. However, existing approaches often face challenges related to inference accuracy and adaptability. To address these issues, we propose ImageSI, a framework integrating deep learning models with semantic interaction techniques for interactive image data analysis. Unlike traditional methods, ImageSI directly incorporates user feedback into the image model, updating underlying embeddings through customized loss functions, thereby enhancing the performance of dimension reduction tasks. We introduce three variations of ImageSI, $\text{ImageSI}_{\text{MDS}^{-1}}$, prioritizing explicit pairwise relationships from user interaction, and $\text{ImageSI}_{\text{DRTriplet}}$ and $\text{ImageSI}_{\text{PHTriplet}}$, emphasizing clustering by defining groups of images based on user input. Through usage scenarios and quantitative analyses centered on algorithms, we demonstrate the superior performance of $\text{ImageSI}_{\text{DRTriplet}}$ and $\text{ImageSI}_{\text{MDS}^{-1}}$ in terms of inference accuracy and interaction efficiency. Moreover, $\text{ImageSI}_{\text{PHTriplet}}$ shows competitive results. The baseline model, WMDS^{-1}, generally exhibits lower performance metrics.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Visual analytics (VA) systems are now essential for navigating complex image datasets in today's data-driven world. These systems enable direct interaction with data representations, allowing users to uncover valuable insights [1]. Semantic interaction (SI) techniques describe a class of interaction methods that aim to enhance VA capabilities by inferring the semantic meaning behind user interactions and adjusting the visualization model according to user feedback [2, 3]. In Andromeda, introduced by Self et al. [4], users can interact with the scatterplot visualization by dragging data points to new positions. This allows them to customize the similarities between points based on their preferences. Through this interaction, called "human spatialization," users can fine-tune the projection according to their preferences and domain knowledge [5]. Dimensionality reduction (DR) methods help VA by simplifying high-dimensional data while retaining important relationships [6, 7]. However, traditional DR methods often struggle to adapt to interactive updates or user-defined similarities [8]. Interactive deep learning (DL) helps to bridge this gap, allowing iteratively feature extraction fine-tuning [5, 9]. By utilizing DL models to extract relevant presentations from image data, interactive DL will enable analysts to refine and explore data visualizations [10, 11].

In previous research, Han et al. developed an interactive DR framework to support SI for exploring image data projections [12]. It uses a Weighted Multi-Dimensional Scaling (WMDS) approach. Their method applies weights to the data space before projection. During SI,

the DR learns new projection weights that best capture the feedback and then applies them for re-projection. Nevertheless, this method faces a critical challenge: limited knowledge retention for iteratively updating a single model. Although the method helps users gradually build a mental model of the underlying data and experiment with various image organizations, it fails to retain knowledge effectively throughout iterative refinement processes. This approach relies on a fixed feature extractor using a pre-trained model, which may overlook dataset nuances if the pre-trained model lacks relevant knowledge about the specific dataset [12]. To address this limitation, I propose to use customized loss backward propagation to optimize the pre-trained model based on user input. Essentially, the difficulty is in learning new information to update the representation and reliably preserve user feedback during the image-sorting process in the feature extractor. Our objective is to develop a methodology that enables the smooth integration of user feedback into the feature extraction process, thereby enhancing the adaptability and effectiveness of interactive DL methods for exploring and understanding complex image datasets.

This work introduces the ImageSI framework to enable interactive DL for analyzing image data. It builds upon prior work by Han et al. [12] and incorporates elements from Self et al.'s Andromeda VA tool [4]. Addressing two key research problems, we first explore methods to fine-tune DL models for SI in image analysis without relying on labeled data or specific features (**RQ1**).

To address this problem, we introduce three distinct approaches. The initial method is inspired by Bian et al.'s DeepSI framework [5], where we replace the pre-trained ResNet-18 model without the fully connected classification layer, represented as ResNet-18_{finetune}, for image data processing. To distinguish this adapted vision for image analysis, we designated it as $\text{ImageSI}_{\text{MDS}^{-1}}$ (customized MDS-based loss).

In the second approach, implemented in the ImageSI framework as $\text{ImageSI}_{\text{DRTriplet}}$ (dimen-

sion reduction with triplet loss), we utilize the ResNet-18$_{\text{finetune}}$ model with triplet loss, accompanied by multidimensional scaling (MDS) to reduce the dimensionality to 2 for projection.

In the third approach, implemented as ImageSI$_{\text{PHTriplet}}$ (projection head with triplet loss) within the ImageSI framework, we employ the ResNet-18$_{\text{projectionhead}}$ model. This model replaces the final fully connected layer of a pre-trained ResNet-18 with a custom sequence of layers. The projection head consists of a sequence of layers, including linear transformation layers and activation functions, designed to map the input features to a 2D space, representing the distances between the anchor, positive, and negative samples in the triplet loss. The linear transformation layers, initially randomize a weight matrix and a bias vector. These layers perform matrix multiplication of the input tensor with the randomized weight matrix and add the bias term. This random initialization can lead to suboptimal initial projection layouts. Therefore, we utilize an autoencoder for the initial projection. The autoencoder is trained to obtain a well-initialized projection layout, which will be used later for updating the projection and comparison. Here, *well-initialized projection layout* refers to a layout where the data points are arranged in such a way that they have a decent adjusted Silhouette score, as detailed in Section 6.1, indicating reasonable separation and clustering in the projected space [13].

Additionally, we introduce triplet samples from coordinate-based triplet loss based on their respective feature vectors in 2D space. This relies only on geometric relationships among feature vectors, eliminating the need for image labels. By utilizing these spatial relationships, the interactive DL framework becomes more versatile and facilitates the exploration and analysis of image datasets without requiring extensive manual annotation.

Second, we aim to assess the inference accuracy and interaction efficiency of ImageSI within the SI loop (**RQ2**). To evaluate its performance, we compare the three proposed Ima-

geSI frameworks, $\text{ImageSI}_{\text{MDS}^{-1}}$, $\text{ImageSI}_{\text{DRTriplet}}$, and $\text{ImageSI}_{\text{PHTriplet}}$, against the baseline model from Han et al., which we will call WMDS^{-1}. Our evaluation strategy comprises two complementary experiments. First, we conduct usage scenarios through case studies utilizing the Animal dataset [14] and Edamame Pods dataset [15]. Second, we perform an algorithm-centered quantitative analysis, involving simulation-based assessments using extended datasets derived from the previously mentioned case studies.

In summary, we claim the following contributions:

- (1): In adapting the $\text{ImageSI}_{\text{MDS}^{-1}}$ framework to analyze image data, we replaced the backbone model with $\text{ResNet-18}_{\text{finetune}}$. This change allowed us to refine feature extraction through fine-tuning, enhancing the framework's semantic interaction (SI) capabilities during iterative sensemaking. Additionally, we integrated the framework with the Andromeda VA system for interactive visualization.

- (2): We improved the SI capabilities in iterative sensemaking using two methods in the ImageSI framework. We used the $\text{ResNet-18}_{\text{finetune}}$ model in $\text{ImageSI}_{\text{DRTriplet}}$ for feature extraction. This model underwent fine-tuning using coordinate-based triplet loss, requiring no labeled data or specific features. This approach reduces dimensionality by integrating MDS. Conversely, for interactive dimensionality reduction in $\text{ImageSI}_{\text{PHTriplet}}$, we used the $\text{ResNet-18}_{\text{projectionhead}}$ model. Similarly fine-tuned with coordinate-based triplet loss, this model replaces MDS with a projection head, while the initial projection is complemented by an autoencoder.

- (3): We conducted two complementary studies comparing the performance of WMDS^{-1}, $\text{ImageSI}_{\text{MDS}^{-1}}$, $\text{ImageSI}_{\text{DRTriplet}}$, and $\text{ImageSI}_{\text{PHTriplet}}$ frameworks. These studies included usage scenarios with two case studies as well as an algorithm-centered quantitative experiment, to assess the inference accuracy and interaction efficiency of our

approach within the SI loop.

Chapter 2

Related Work

2.1 ResNet

In the field of deep learning, especially in computer vision, ResNet (Residual Neural Network) is a landmark architecture. ResNet created skip connections, also referred to as shortcuts, to address the problem of disappearing gradients in deep neural networks. These connections allow for a more direct gradient flow during training [16].

The residual block, which consists of rectified linear unit (ReLU) activation functions and batch normalization after convolutional layers, is a key part of ResNet [17]. The innovation component is in the residual connections, which let gradients add the input directly to the block's output rather than passing through one or more convolutional layers [16]. ResNet can learn residual mappings thanks to this mechanism, which facilitates the efficient training of deeper networks [18].

ResNet architectures are available in different depths based on the number of layers and residual blocks; these are indicated by suffixes such as ResNet-18, ResNet-34, and ResNet-50. Usually, these architectures are pre-trained on extensive image datasets like ImageNet [19], where they demonstrate exceptional performance in identifying significant features from images.

We utilize ResNet's capabilities in our framework to process image data. For feature ex-

traction fine-tuning, we use the ResNet-18_{finetune} model that we integrate into the ImageSI pipeline. Recent research demonstrates the efficacy of fine-tuning off-the-shelf ResNet-50 weights for iris recognition tasks, showing improved accuracy compared to both off-the-shelf weights and models trained from scratch [20]. Our approach uses customized loss backward propagation, which is different from traditional fixed feature extractors [12] and makes it easier to optimize the ResNet-18_{finetune} model according to user feedback. We improve the ResNet model to more accurately capture pertinent features specific to the examined image dataset through interactive user interactions.

2.2 Dimensionality Reduction Techniques

In data analytics, dimensionality reduction techniques (DRTs) have become essential for handling the complexity of high-dimensional datasets across various domains. These methods help transform data from a high-dimensional space to a lower-dimensional representation while trying to maintain important information and patterns present in the original data [21]. The many useful benefits that DRTs provide are what drive their use. First, by reducing the number of dimensions and easing the load on data storage and computational processing, these techniques improve the efficacy of analyses. Second, they help to mitigate the negative effects of redundant, irrelevant, or noisy data, thereby enhancing data quality and enabling more accurate analyses. Additionally, DRTs solve the problem of visualizing high-dimensional data by enabling data visualization in lower-dimensional spaces [22].

Dimensionality reduction typically involves two main strategies: feature extraction and feature selection. Feature extraction is the process of converting the initial features into a new set while retaining essential information, while feature selection entails choosing a subset of the original features and eliminating those that are unnecessary or redundant [23].

Linear Dimension Reduction Techniques (LDRTs) represent a fundamental category of dimensionality reduction methods that employ linear transformations to map high-dimensional data to lower-dimensional spaces [24]. Principal Component Analysis (PCA) is a renowned LDRT designed to identify orthogonal components that capture the maximum variance within the data [25]. PCA is widely used in many fields, such as robotics, exploratory data analysis, and image and speech processing.

while LDRTs are commonly used for dimensionality reduction, nonlinear methods provide enhanced capacities for representing intricate data structures. Autoencoders, a type of deep learning model, exemplify nonlinear DRTs by avoiding the assumption of linearity in the data. Autoencoders are more flexible in capturing complex data patterns because they use neural networks to learn how to encode high-dimensional data into lower-dimensional representations [26]. Among the nonlinear DRTs, t-Distributed Stochastic Neighbor Embedding (t-SNE) and Uniform Manifold Approximation and Projection (UMAP) have gained considerable attention for their efficacy in preserving both local and global data structures [27].

t-SNE employs random walks on neighborhood graphs to visualize the structure of extensive datasets. This helps to overcome the tendency to cluster points together and generate high-quality visualizations. Conversely, UMAP creates a topological representation of high-dimensional data using fuzzy simplicial set representations and local manifold approximations, which is particularly effective in maintaining both local and global relationships within the data [28].

In our study, we use MDS as a nonlinear DRT. Previous research has employed WMDS^{-1}, as an interactive DR method, to learn new weights for WMDS based on the interaction such that the pairwise distances in the weighted feature space are proportional to the pairwise distances of the manipulated points. Our approach simplifies the process by eliminating the weighting scheme inherent in WMDS^{-1}. Instead of relying on weights, our method preserves

user feedback through the backpropagation of a DL model, contrasting with the previous method. Additionally, we deliberately choose MDS over t-SNE and UMAP to ensure a fair comparison of $\text{ImageSI}_{\text{MDS}^{-1}}$, $\text{ImageSI}_{\text{DRTriplet}}$, and $\text{ImageSI}_{\text{PHTriplet}}$ against the baseline model WMDS^{-1}, which employs Weighted MDS.

2.3 Neural Network Projections

Compared to traditional DRTs, Neural Network Projections (NNPs) offer a revolutionary method of data projection. NNPs use the power of neural networks to generate low-dimensional representations of high-dimensional data, in contrast to techniques like PCA or t-SNE. This innovative paradigm shift in data projection arises from the limitations of conventional techniques to adequately capture the intricate structures found in modern high-dimensional datasets [29].

Traditional DRTs like PCA often lack the flexibility to capture the complex nonlinear relationships present in deep neural network representations [30]. As datasets grow more high-dimensional, traditional methods may fail to preserve important features and relationships, resulting in deficient projections. Additionally, the effectiveness of traditional methods in capturing the underlying data structure can be limited if they depend on predefined distance metrics or assumptions about data distribution [31, 32].

In contrast, NNP methods for data projection within neural networks, such as autoencoders and DrLIM, offer several advantages [29, 30, 31]. Autoencoders aim to generate compressed, low-dimensional representations by training neural networks to reconstruct their inputs [29]. Autoencoders can capture complex nonlinear relationships in the data by utilizing the expressive capacity of neural networks, resulting in more meaningful and accurate projections. Similarly, DrLIM learns a globally coherent nonlinear function that maps high-dimensional

data to a low-dimensional manifold, providing an alternative to the constraints of traditional methods [31]. DrLIM can handle complex data transformations and does not rely on predefined distance metrics. Moreover, NNP methods offer flexibility and scalability, enabling parallelization and out-of-sample functionality [29]. This enables efficient processing of large-scale datasets and integration into interactive systems for data exploration and sensemaking. González Martínez et al. explore the use of neural encoders to mimic the dimensionality reduction behavior of non-invertible algorithms like UMAP, NNP methods aim to learn low-dimensional embeddings directly from the data using multilayer neural networks [33].

We take advantage of the benefits of Neural Network Projections (NNPs) in the $\text{ImageSI}_{\text{PHTriplet}}$ framework by training an autoencoder for initial projection and adding a projection head to map the dimensions to 2.

2.4 Visual Analytics with Andromeda

VA is a powerful approach that combines statistical models, visualization techniques, and human intuition to understand large and complex datasets [34]. It allows users to interactively explore and analyze data, helping with sensemaking through filtering, gathering, synthesizing, and forming hypotheses [35, 36].

One key aspect of VA is interaction, which allows users to manipulate visualizations based on their domain expertise, thus guiding the analysis process. SI introduces a novel approach to interaction within the VA context. It enables users to interact directly within the spatial metaphor of visualizations, thereby tightly coupling user actions with underlying statistical models [34].

Andromeda is a VA tool developed by Self et al. [4]. Using Visual to Parametric Interaction (V2PI), Andromeda enables users to interact with data points in visualizations, impacting the analysis process. V2PI simplifies the mathematical models, allowing users to concentrate on exploring data according to their expertise and hypotheses [37].

Andromeda utilizes WMDS to spatialize high-dimensional data into two dimensions, where distance reflects relative similarity [4]. Through the Andromeda interface, users can adjust spatializations using both parametric (PI) and observation-level interactions (OLI). The tool's Object View displays the resulting object layout calculated by WMDS, allowing users to explore data points and manipulate them directly to provide input to the algorithm [4, 38].

In the Andromeda system, users can use the Parameter View to adjust the importance of different dimensions by interacting with lines. These interactions cause the layout to update dynamically, with the WMDS algorithm recalculating the layout in real-time based on the new weight vector [4].

Within our framework, we incorporated only the OLI view into our adapted Andromeda visual analytics tool. This choice is based on our limited comprehension of the representation of features extracted by ResNet. Although we could include PI in Andromeda and allow users to directly manipulate model parameters, this would result in users being uninformed about each feature parameter. Therefore, by concentrating on OLI, our goal is to enhance communication with underlying models without overwhelming users with complicated details that are unclear due to our limited understanding of feature representations.

Chapter 3

Background

In this chapter, we present an overview of two prominent frameworks, WMDS^{-1} [12] and DeepSI [5]. The former serves as a benchmark for comparison with the three proposed ImageSI frameworks, while the latter provides background to contextualize the discussion of the proposed ImageSI$_{\text{MDS}^{-1}}$ framework.

3.1 WMDS^{-1}

WMDS^{-1} is a novel framework designed to ease human-in-the-loop sensemaking through interactive visualization, leveraging features extracted from a pre-trained ResNet model. Their proposed pipeline uses WMDS to weight deep learning image features, allowing interaction through direct manipulation of points in the 2D plot and learning new weights through WMDS^{-1}, a technique first introduced in Andromeda [4], to enable interactive DR.

WMDS^{-1} learns new weights for WMDS based on the interaction such that the pairwise distances in the weighted feature space are proportional to the pairwise distances of the manipulated points. Specifically, it computes new weights optimized for preserving specified relationships after the analyst repositions a subset of points, which is represented as y^*. This process effectively captures human feedback. WMDS^{-1} updates the weights using the following formula:

$$w = \underset{w_1,\ldots w_d}{\arg\min} \sqrt{\frac{(\sum_{i<j\leq N}(d_L(y_i^*, y_j^*) - d_H(w, x_i, x_j))^2}{\sum_{i<j\leq N} d_H(w, x_i, x_j)^2}} \tag{3.1}$$

Through the user interactions, this equation yields a vector of dimension weights that most closely adhere to the 2D pairwise similarities. By normalizing the weight vector to sum to 1, the high-dimensional distances are brought into a roughly constant-sized space. To create a layout that takes the analyst's input into account, the images are then re-projected using the updated weights.

3.2 DeepSI

Using deep learning representations, DeepSI is a state-of-the-art framework for human-in-the-loop sensemaking that enables interactive visualization. Unlike WMDS^{-1}, it incorporates user SI to fine-tune the model and update the subsequent high-dimensional embeddings to align with user intentions, rather than weighting the data space before projection. To facilitate user interaction, the framework includes both forward model projection and backward model-updating directions.

The framework starts the forward direction by using the BERT model's forward propagation calculation to create new representations x for the dataset. In this procedure, the dataset is subjected to the BERT model with the current BERT parameters w_{BERT}. Mathematically, this is represented as:

$$x = \text{BERT}(d, w_{\text{BERT}}) \tag{3.2}$$

where d represents the dataset and w_{BERT} denotes the current BERT parameters.

Next, MDS is used to project the high-dimensional deep learning representations onto a 2D spatialization. The objective function of MDS is to minimize the discrepancy between the distances of the data points in low-dimensional and high-dimensional spaces, which helps to guide the projection. The expression for the objective function is:

$$\mathbf{y} = \arg\min_{y} \sum_{i<j\leq N} (\text{dist}_L(y_i, y_j) - \text{dist}_H(x_i, x_j))^2 \tag{3.3}$$

Here, $\mathbf{y}$ represents the 2D spatialization, dist_L denotes the low-dimensional distance function, and x_i represents the i-th element of the high-dimensional representations of the dataset. In contrast to WMDS^{-1}, the fine-tuned representation x itself determines the updates to $\mathbf{y}$ rather than the high-dimensional distance function being explicitly weighted.

In the backward direction, analysts adjust the visual arrangement by relocating specific samples to represent desired similarities. Introduced by Bian et al., MDS^{-1} is a loss function that aligns the pairwise distances in the embedding space with those specified in the 2D space, allowing for the adjustment of model weights to enhance image embeddings. This adjustment process uses the similarities defined by humans on the relocated data points to fine-tune the BERT model parameters w_{BERT} and enhance the high-dimensional representations x. In simpler terms, the objective is to modify the BERT weights w_{BERT} to minimize the difference between distances in lower and higher dimensions of data points adjusted through backpropagation:

$$w_{\text{BERT}} = \arg\min_{w_{\text{BERT}}} \sum_{i<j\leq n} (\text{dist}_L(y_i, y_j) - \text{dist}_H(\text{BERT}(d_i, w_{\text{BERT}}), \text{BERT}(d_j, w_{\text{BERT}})))^2 \tag{3.4}$$

By using a collaborative process involving human feedback, the BERT model can be adjusted to create customized representations tailored to individual users and specific tasks. This enables the model to accurately capture the exact intentions of analysts.

Chapter 4

Methodology and Workflows

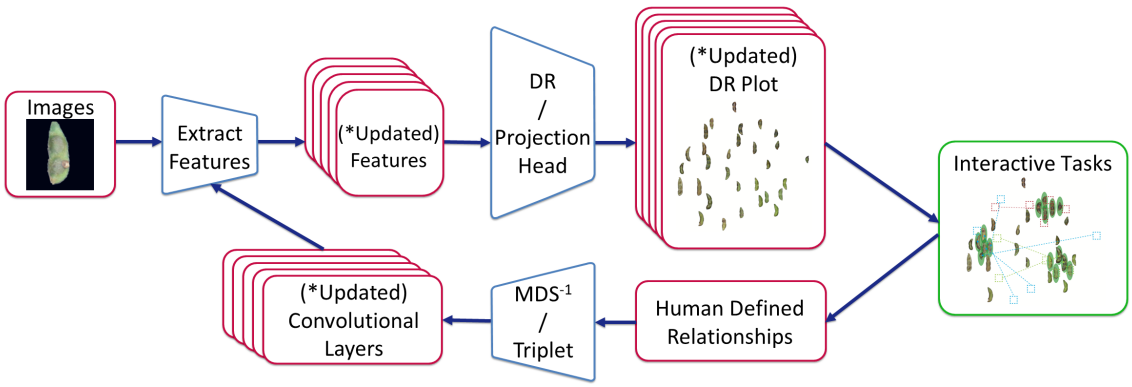


Figure 4.1: The ImageSI pipeline overview illustrates the process: initially, features are extracted using a fine-tuned pre-trained ResNet-18 model. These features are then inputted into a DR method, MDS, or a projection head to reduce dimensionality to 2, resulting in a 2D projection. The human interaction loop returns newly defined relationships to the fine-tuned ResNet-18 model following interactions. SIs facilitate model fine-tuning interactively through backpropagation using metric or triplet loss. Subsequently, the fine-tuned ResNet model with updated Convolution layers extracts updated features to capture the analyst's intent. The interaction loop can be iterated at desired times.

In this section, we outline the anticipated user workflow and the basic approach used. Figure 4.1 illustrates the process flow.

4.1 ImageSI Framework

This section covers the initialization process, referred to as the Initial State, and the interactive functionalities of the ImageSI framework, $\text{ImageSI}_{\text{MDS}^{-1}}$, $\text{ImageSI}_{\text{DRTriplet}}$, and

$\text{ImageSI}_{\text{PHTriplet}}$.

4.1.1 Initial State

ImageSI starts by initializing datasets and loading image data using a data loader for subsequent processing tasks. Then, we load both the $\text{ResNet-18}_{\text{finetune}}$ and $\text{ResNet-18}_{\text{projectionhead}}$ models to fine-tine feature extraction.

Subsequently, we iterate through the images in the data loader, using the instantiated models to extract features and aggregate them into a dataframe with corresponding image indices. We then employ MDS to reduce the data's dimensionality, creating a 2D projection.

To ensure consistency across executions, we further process the 2D projection by rotating it via Principal Component Analysis (PCA) and aligning the first principal component with the y-axis.

For initializing $\text{ImageSI}_{\text{DRTriplet}}$ and $\text{ImageSI}_{\text{MDS}^{-1}}$, the process follows the same steps as described above. However, for $\text{ImageSI}_{\text{PHTriplet}}$, we performed an extra step to use an autoencoder trained to obtain an initialized projection layout. This initialized layout will be used for both comparison and projection updates in the future.

4.1.2 ImageSI Interaction

The interaction design of ImageSI incorporates elements from the existing VA system, Andromeda, to facilitate SI. SI interprets the underlying meaning behind user interactions with visualizations. By understanding user intents and analytical reasoning, SI adjusts the visualization model to better align with user needs and preferences. In the ImageSI interface, users can engage in SI by directly modifying the layout of the scatterplot visualization, thereby

influencing the projected data points based on their semantic understanding and analytical insights.

When we look at the model's projections, the scatterplot shows how data points are similar to each other according to the ResNet-based model, showcasing how the model visualizes their relationships in space. Users can observe how the model interprets similarities between data points based on their spatial arrangement on the scatterplot.

Users can manipulate the scatterplot by relocating data points in the backward model-updating direction. This enables them to tailor the similarities between points to suit their preferences. Termed "human spatialization," this interaction empowers users to refine the projection according to their preferences [5].

This bidirectional interaction fosters direct and effective communication between the underlying model and human analysts. Users can refine the projection layout based on their insights, effectively teaching the model new relationships between data points.

Users can interactively adjust the projection plot by moving points within the 2D space from their original positions. An analyst might, for instance, project a series of images showing various animal features, such as open-mouth and closed-mouth. But it's possible that the original projection didn't distinguish between these features. The analyst tells the model that the representative images of each type are different and should be arranged differently by selecting and dragging those images to opposite positions.

Once the analyst completes their interaction, ImageSI iteratively fine-tunes the ResNet based on the learned information from the user interaction, optimizing the model to create a projection layout that best reflects the specified similarities.

4.2 Model Description

This section outlines the model design and pipeline, including implementation details of the ImageSI framework. Three distinct approaches are outlined:

$\text{ImageSI}_{\text{MDS}^{-1}}$ utilizes the $\text{ResNet-18}_{\text{finetune}}$ model from Bian et al.'s DeepSI framework, replacing the fully connected classification layer. It incorporates a customized MDS-based loss.

$\text{ImageSI}_{\text{DRTriplet}}$ employs triplet loss and multidimensional scaling (MDS) for dimensionality reduction to 2 for projection.

$\text{ImageSI}_{\text{PHTriplet}}$ uses the $\text{ResNet-18}_{\text{projectionhead}}$ model, where the final fully connected layer is replaced with a custom projection head. The projection head consists of linear transformation layers and activation functions designed to map input features to a 2D space, representing distances in the triplet loss.

4.2.1 Model Design

To address the research problem, we propose a primary design goal for the ImageSI framework: Adapt DL models for SI in image analysis without requiring labeled data or specific features.

Traditional methods for learning embeddings, such as the triplet loss, require labeled data to annotate the anchor, positive, and negative samples [39]. However, relying on labels incurs extensive human annotations, increasing costs and time consumption. Moreover, labeled data might not always be available or adequately represent the variability present in the dataset [40, 41].

We introduce the coordinate-based triplet loss to address these challenges. Unlike traditional methods, the coordinate-based approach operates directly on the spatial coordinates of data points, eliminating the need for labeled data. Both ResNet-18_{finetune} and ResNet-$18_{\text{projectionhead}}$ models use this loss function to extract meaningful representations from the spatial coordinates of data points. In the embedding space, the models learn to distinguish dissimilar data points and cluster closely similar ones using triplets of anchor, positive, and negative samples. This approach ensures the models' flexibility for different tasks and datasets without relying on labeled data.

Chapter 5

Model Pipeline

The model pipeline comprises several key components that collaborate to train the ResNet model for image feature extraction and refinement. The process entails forward and backward steps, encompassing feature extraction using ResNet, triplet sample selection, triplet loss calculation, and model training. The pipeline is structured as follows:

5.1 ResNet-18$_{\text{finetune}}$

Based on the ResNet-18 architecture, the model ResNet-18$_{\text{finetune}}$ is intended for feature extraction tasks that require fine-tuning. The model loads the ResNet-18 model that has already been trained and removes its last fully connected layer during initialization, leaving only the feature extraction layers [42]. Through fine-tuning the remaining layers to extract task-specific features from the input data, this modification enables the model to adapt to new tasks. During the forward pass, the modified ResNet-18 model processes the input data to produce feature representations that can be utilized for different downstream tasks.

5.2 ResNet-18$_{\text{projectionhead}}$

ReNet-18$_{\text{projectionhead}}$ is built using a pre-trained ResNet-18 model's architecture. It replaces the final fully connected layer with a custom sequence of layers, consisting of two linear

transformation layers followed by Leaky ReLU activation functions. The initial linear layer reduces the dimensionality of the feature space from the original size of the fully connected layer to 512, while the subsequent linear layer further reduces it to 10 dimensions. Additionally, a separate sequence of layers is developed to compute the triplet loss, involving a linear layer that projects the 10-dimensional feature vector into a 2-dimensional space. During the forward pass, input data is passed through the pre-trained ResNet-18 model to extract features, which are then processed by the triplet loss layer to calculate positive and negative distances.

Mathematically, given an input image tensor x, the ResNet-$18_{\text{projectionhead}}$ computes the feature vector $\mathbf{f}$ using the pre-trained ResNet-18 model as follows:

$$\mathbf{f} = \text{ResNet}(x)$$

Here, ResNet(x) represents the process of passing the input image tensor x through the model, resulting in the extraction of relevant features represented by the feature vector $\mathbf{f}$.

The feature vector $\mathbf{f}$ is then passed through the Triplet Loss layer to obtain the final 2-dimensional embeddings, denoted as $\mathbf{e}$:

$$\mathbf{e} = \text{Triplet\_Loss}(\mathbf{f})$$

In this equation, Triplet_Loss($\mathbf{f}$) refers to the operation performed by the Triplet Loss layer on the feature vector $\mathbf{f}$, resulting in the generation of 2-dimensional embeddings $\mathbf{e}$.

5.2.1 Autoencoder

The encoder and decoder in our implementation of the autoencoder are both made up of linear transformation layers. The input data is compressed by the encoder into a lower-dimensional representation during the forward pass, and the original input is reconstructed by the decoder from this encoded representation.

We use the autoencoder to obtain an initial projection layout for $\text{ImageSI}_{\text{PHTriplet}}$. We first extract features from the pre-trained ResNet-18 model and then run these features through the encoder to obtain the encoded representation. As the first projection layout, this encoded representation captures important details about the input data in a 2D space.

5.3 Triplet Sample Selection

Triplet samples are selected from the dataset to facilitate triplet learning. Each triplet comprises an anchor sample, a positive sample (similar to the anchor), and a negative sample (dissimilar to the anchor). ImageSI forms triplets, using each point manipulated in the interaction as an anchor. Triplets of images are chosen such that, for a given anchor, ImageSI generates a pool of positive and negative samples by calculating the absolute differences between the anchor coordinates and the coordinates of all other points. To identify a pool of positive samples, it calculates the Euclidean distance between the anchor and each moved point and sets a threshold for the minimum absolute difference such that any point below this threshold is put into the pool. For a given triplet at that anchor, the positive sample is then randomly selected from this pool. Negative samples are identified similarly, with points having a maximum absolute difference exceeding the threshold considered as negative samples. This process generates a pool of positive and negative samples for each anchor,

used to form the triplets for training.

The anchor, positive, and negative samples are denoted as x. The Euclidean distance between two feature vectors x_1 and x_2 is calculated as:

$$\text{distance}(x_1, x_2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2}$$

5.4 Triplet Loss Calculation

The triplet loss calculation quantifies the similarity between anchor-positive pairs and the dissimilarity between anchor-negative pairs. It encourages the model to learn feature representations such that the positive samples are closer to the anchor samples than the negative samples. The Triplet Loss L_{triplet} is computed as:

$$L_{\text{triplet}} = \frac{1}{N} \sum_{i=1}^{N} \max(d(a_i, p_i) - d(a_i, n_i) + \alpha, 0)$$

where $d(x_1, x_2)$ represents the Euclidean distance between x_1 and x_2, a_i, p_i, and n_i represent the feature vectors of the anchor, positive, and negative samples respectively for the ith triplet, N represents the overall count of triplets, while α denotes the margin value, a hyperparameter controlling the minimum difference required between positive and negative distances.

5.5 Model Training

The model is trained using triplet samples selected from the dataset, to minimize the Triplet Loss. Each training epoch involves forwarding the input images through the ResNet-based model, computing gradients, and updating model parameters through backward propagation. An optimizer is utilized to update the model parameters using the calculated gradients.

For $\text{ImageSI}_{\text{DRTriplet}}$ and $\text{ImageSI}_{\text{PHTriplet}}$, it's recommended to use a learning rate of 1×10^{-5} for optimal performance. The number of epochs needed depends on the complexity of the interactive tasks. More challenging tasks require more epochs, while simpler tasks require fewer.

The training process continues for a specified number of epochs set by users, with the option of early stopping based on the consecutive epochs with zero loss. If the specified threshold for consecutive zero-loss epochs is reached, the training process terminates early to prevent overfitting and ensure efficient model convergence.

Through this iterative training pipeline, the model gradually learns to extract meaningful feature representations from the image data.

Chapter 6

Usage Scenarios

In this section, we demonstrate the practical applications of our framework by comparing the three proposed ImageSI frameworks, $\text{ImageSI}_{\text{MDS}^{-1}}$, $\text{ImageSI}_{\text{DRTriplet}}$, and $\text{ImageSI}_{\text{PHTriplet}}$, against the baseline model: WMDS^{-1}. We evaluate their performance across two real-world scenarios involving four image-sorting tasks. As part of our analysis, we incorporate an evaluation of the projected layouts, utilizing the adjusted Silhouette score to evaluate the clustering quality in each scenario.

6.1 Projection Evaluation

In evaluating the effectiveness of our method in capturing simulated user interaction, we utilize the adjusted Silhouette score [13]. The metric evaluates clustering quality by considering the tightness of points within clusters (cohesiveness) and the distinctiveness between clusters (separation). The Silhouette score, after adjustment, spans from -2 to 2. Near-zero scores suggest cluster overlap, negatives imply misassignments, and positives indicate well-separated clusters [43].

Following Han et al.'s approach [12], we are creating a DR technique that considers human feedback. Instead of focusing only on compact and well-separated clusters, we value the meaningful insights found in the dispersion of clusters. We understand that important insights can be hidden in how clusters are spread. This is why our target Silhouette score

is around 0.5, preferring arrangements where data points are typically twice as far from the closest class as they are from their own class. To emphasize this preference, we adjust the Silhouette score by doubling it so that an ideal score is one. Scores below one signal too much spreading out, while scores above one suggest excessive clustering.

The adjusted Silhouette score is calculated using the following formula:

$$\text{Adjusted Silhouette Score} = \frac{b - a}{\max(a, b)} \times 2$$

where a represents the mean intra-cluster distance, which measures the average distance between a sample and other samples within the same cluster. b represents the mean nearest-cluster distance, which calculates the average distance between a sample and samples in the nearest cluster that the sample does not belong to.

A higher adjusted Silhouette score signifies better clustering performance, with values nearing 1 suggesting well-separated clusters, while those nearing -1 imply potential misassignments. Scores near 0 imply overlapping clusters.

6.2 Case Study: Animals

In this case study, we investigate the capabilities of our framework, $\text{ImageSI}_{\text{MDS}^{-1}}$, $\text{ImageSI}_{\text{DRTriplet}}$, and $\text{ImageSI}_{\text{PHTriplet}}$, alongside the baseline framework, WMDS^{-1}, in handling various sorting tasks using two distinct animal image datasets.

For the first task, we utilize an animal dataset obtained from Kaggle, consisting of 20 images of sharks and snakes [14]. This dataset aims to evaluate the model's ability to distinguish between animals based on the status of their mouths. Specifically, the dataset includes

5 images each of open-mouth sharks, open-mouth snakes, closed-mouth sharks, and closed-mouth snakes, providing a balanced representation of both open and closed-mouthed animals for accurate sorting.

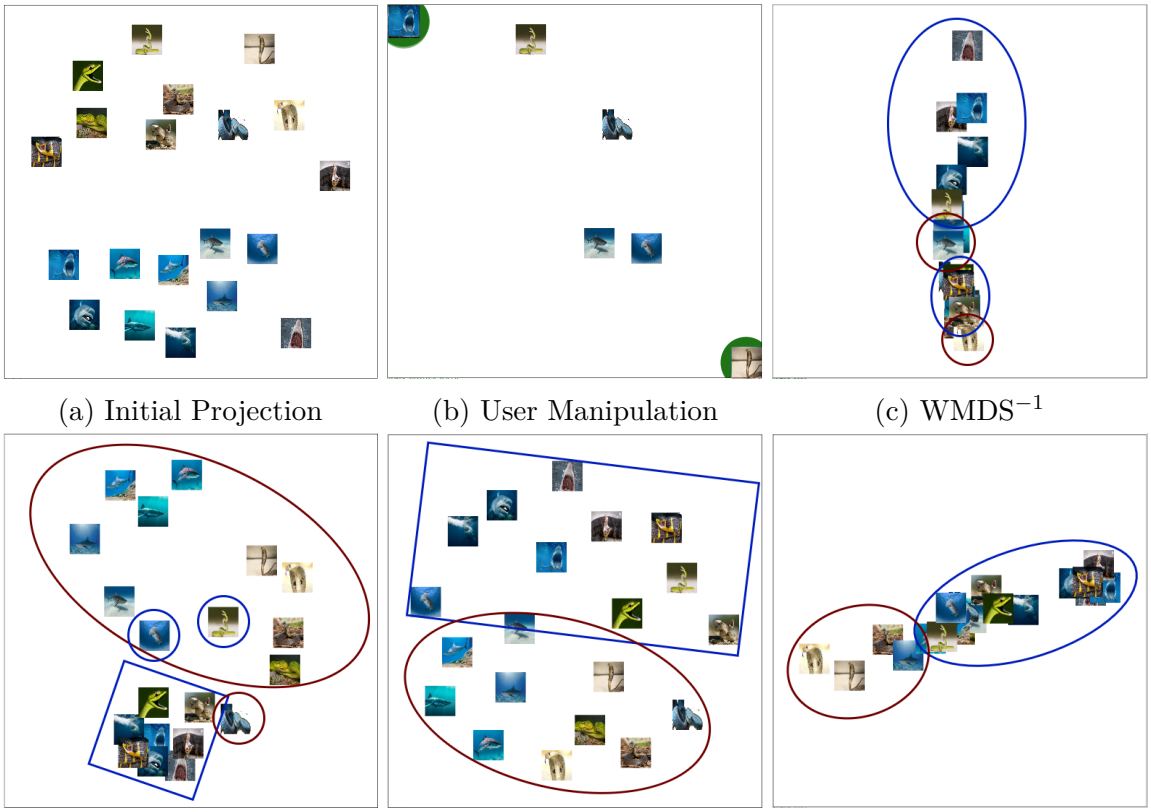
For the second task, we employ a mixed dataset comprising 30 images distributed across three classes: "single horse," "human and horse," and "multiple horses." These images were sourced from Kaggle [14]. This dataset presents the model with the challenge of distinguishing between various contexts involving horses, including scenarios with humans alongside the horses and images featuring multiple horses. This diverse range of contexts presents a challenging sorting task that requires the model to understand and differentiate between different visual situations involving horses and humans.

6.2.1 Task 1: Open-Mouth and Closed-Mouth Animals

We start by loading a dataset comprising images of sharks and snakes, as depicted in Figure 6.1(a). Upon inspection, we note that some animals in the images have open mouths while others have closed mouths.

To convey this information to the model, we engage in interactive manipulation of the projection by segregating images of animals with open mouths from those with closed mouths on three frameworks. As shown in Figure 6.1(b), we selected 8 animals with open mouths (4 sharks and 4 snakes) and 8 animals with closed mouths (4 sharks and 4 snakes). We positioned the 8 animals with open mouths in the top left corner and the 8 animals with closed mouths in the bottom right corner to visually highlight their distinctions. Table 6.1 shows the adjusted Silhouette scores for all frameworks.

Figure 6.1(c) exhibits the updated projection for WMDS^{-1}. Blue contours represent animals with open mouths, while red contours represent animals with closed mouths. The



(a) Initial Projection (b) User Manipulation (c) WMDS^{-1}

(d) $\text{ImageSI}_{\text{MDS}^{-1}}$ (e) $\text{ImageSI}_{\text{DRTriplet}}$ (f) $\text{ImageSI}_{\text{PHTriplet}}$

Figure 6.1: Usage scenario on the Open-Mouth vs. Closed-Mouth Animals dataset: (a-f) shows the procedure for conducting exploratory analysis on 20 images. In (a), we show the initial projection. In (b), users separate the images of "open-mouth animals" from those of "closed-mouth animals" by dragging them apart, highlighting the distinct "mouth" feature. (c) shows the updated projection for WMDS^{-1}, where blue contours include open-mouth animals and red contours include closed-mouth animals. Similarly, (d) shows the updated projection for $\text{ImageSI}_{\text{MDS}^{-1}}$, (e) for $\text{ImageSI}_{\text{DRTriplet}}$, and (f) for $\text{ImageSI}_{\text{PHTriplet}}$.

updated projection reveals cluster overlap, with two blue and two red contours indicating misprojections. WMDS^{-1} yields the lowest adjusted Silhouette score among the evaluated frameworks.

In Figure 6.1(d), we see the updated projection for $\text{ImageSI}_{\text{MDS}^{-1}}$. One closed-mouth animal and two open-mouth animals are misclassified. Open-mouth animals predominantly group at the bottom, while closed-mouth animals cluster in the middle right and top portions.

Figure 6.1(e) demonstrates the updated projection for $\text{ImageSI}_{\text{DRTriplet}}$. A distinct separation is observed, with open-mouth animals at the top and closed-mouth animals at the bottom.

Finally, Figure 6.1(f) depicts the updated projection for $\text{ImageSI}_{\text{PHTriplet}}$. Despite a slightly higher adjusted Silhouette score compared to $\text{ImageSI}_{\text{DRTriplet}}$, around five closed-mouth images and the open-mouth animals overlap, leading to dense clustering.

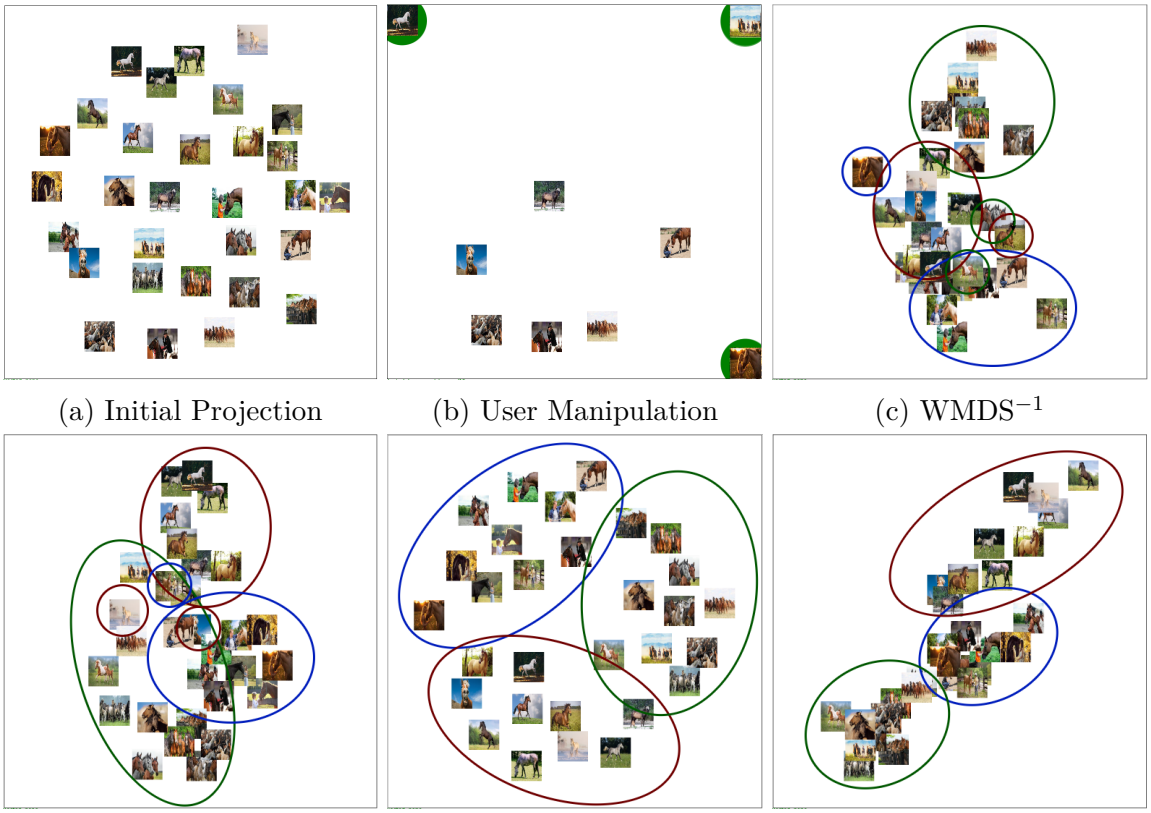
Table 6.1: Adjusted Silhouette Scores for Open-mouth vs. Closed-mouth Animals Dataset

Framework	**Adjusted Silhouette Score**
WMDS^{-1}	0.458
$\text{ImageSI}_{\text{MDS}^{-1}}$	0.679
$\text{ImageSI}_{\text{DRTriplet}}$	0.653
$\text{ImageSI}_{\text{PHTriplet}}$	0.683

6.2.2 Task 2: Single Horse, Human and Horse, and Multiple Horses

We load a mixed dataset containing images of humans and horses. Initially, the projection groups similar images. However, upon closer inspection, we notice that some images contain a single horse, some contain a human and a horse, and others contain multiple horses, as depicted in Figure 6.2(a).

To address this, we first employ interactive manipulation of the projection to separate images



(a) Initial Projection (b) User Manipulation (c) WMDS^{-1}

(d) $\text{ImageSI}_{\text{MDS}^{-1}}$ (e) $\text{ImageSI}_{\text{DRTriplet}}$ (f) $\text{ImageSI}_{\text{PHTriplet}}$

Figure 6.2: Usage scenario on the Single Horse vs. Human and Horse vs. Multiple Horses dataset: (a-f) depict the process for conducting exploratory analysis on 30 images. In (a), we present the initial projection. In (b), the user adjusts the projection by segregating the images featuring "human and horse," and "multiple horses" from those showing only "single horse." This highlights the presence of the "human" object and the concept of "multiple" features. The updated projection, (c), showcases the output for WMDS^{-1}, where red contours represent single horses, blue contours represent humans and horses, and green contours represent multiple horses. Similarly, (d) displays the updated projection for $\text{ImageSI}_{\text{MDS}^{-1}}$, (e) for $\text{ImageSI}_{\text{DRTriplet}}$, and (f) for $\text{ImageSI}_{\text{PHTriplet}}$.

containing humans from those with only animals. By dragging images of humans and horses apart from single horse images, we guide the model to learn about human objects. Then, we separate the multiple horses from the single horse by dragging them, teaching the model to recognize scenarios with multiple objects. The user manipulation is depicted in Figure 6.2(b). We selected 8 images with single horses, 8 with humans and horses, and 8 with multiple horses. We positioned the single horse images in the top left corner, multiple horses in the top right corner, and human and horse images in the bottom right corner to visually highlight their distinctions. Table 6.2 shows the adjusted Silhouette scores for all frameworks.

Figure 6.2(c) demonstrates the updated projection for WMDS^{-1}. Red contours depict single horses, blue contours represent humans and horses, and green contours represent multiple horses. WMDS^{-1} yields the lowest adjusted Silhouette score among the evaluated frameworks, indicating insufficient separation of clusters, as evidenced by two red, two blue, and two green contours.

In Figure 6.2(d), we see the updated projection for $\text{ImageSI}_{\text{MDS}^{-1}}$, which also exhibits two blue and two red contours, indicating a failure to differentiate between single horses and images containing humans and horses. Additionally, there are instances of overlapping images.

Figure 6.2(e) presents the updated projection for $\text{ImageSI}_{\text{DRTriplet}}$, revealing a distinct separation with all three clusters.

Finally, Figure 6.2(f) illustrates the updated projection for $\text{ImageSI}_{\text{PHTriplet}}$, boasting the highest adjusted score among the frameworks. However, some images containing both humans and horses, as well as multiple horse images, appear densely clustered, impacting clarity.

Table 6.2: Adjusted Silhouette Scores for Single Horse vs. Human and Horse vs. Multiple Horses Dataset

Framework	**Adjusted Silhouette Score**
WMDS^{-1}	0.302
$\text{ImageSI}_{\text{MDS}^{-1}}$	0.667
$\text{ImageSI}_{\text{DRTriplet}}$	0.890
$\text{ImageSI}_{\text{PHTriplet}}$	0.923

6.3 Case Study: Edamame Pods

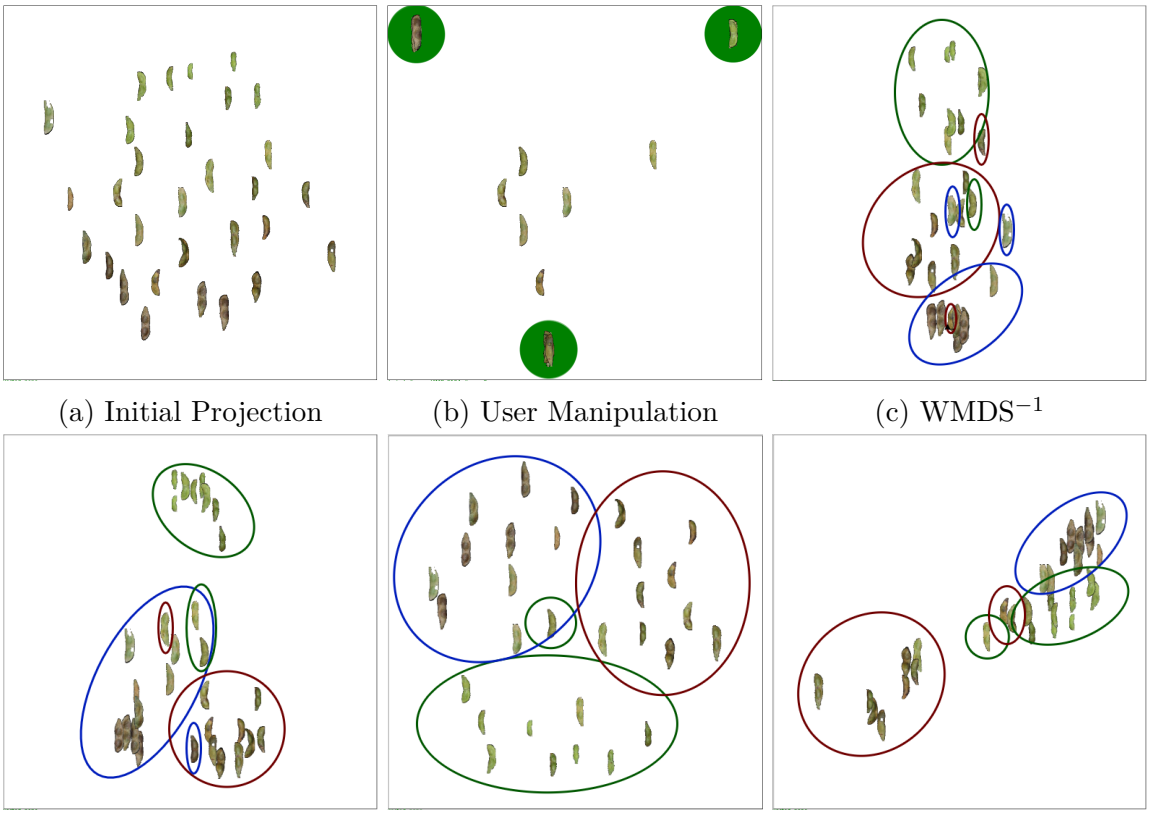
In this case study, we examine two datasets containing images of edamame pods, aiming to gain insights into the characteristics associated with pod maturity stages and the number of seeds per pod.

The datasets are sourced from the Plant Sciences Department at Virginia Tech [15]. The first dataset consists of 30 images of edamame pods, each containing two seeds. These images are categorized into three maturity stages: "ready-to-harvest," "late-to-harvest," and "diseased," with 10 images in each category.

The second dataset also contains 30 images of edamame pods, categorized based on the number of seeds present: "one-seed," "two-seeds," and "three-seeds." Similar to the first dataset, each category comprises 10 images. Importantly, these images cover all three maturity stages, providing a comprehensive dataset for exploring the relationship between maturity stages and seed counts in edamame pods.

6.3.1 Task 1: Maturity Stage

For the first task, our focus lies on organizing the edamame pod images based on their maturity stages, while disregarding the noise introduced by the number of seeds. We hypothesize



(a) Initial Projection (b) User Manipulation (c) WMDS^{-1}

(d) $\text{ImageSI}_{\text{MDS}^{-1}}$ (e) $\text{ImageSI}_{\text{DRTriplet}}$ (f) $\text{ImageSI}_{\text{PHTriplet}}$

Figure 6.3: Usage scenario on the Ready-to-Harvest vs. Late-to-Harvest vs. Diseased Edamame Pods dataset: (a-f) demonstrate the exploratory analysis process on 30 images. In (a), the initial projection is shown. In (b), the user manipulates the projection by segregating the "ready-to-harvest" and "late-to-harvest" images from the "diseased" images, emphasizing the concept of "maturity" features. The updated projection, (c), showcases the output for WMDS^{-1}, where red contours represent diseased pods, blue contours represent late-to-harvest pods, and green contours represent ready-to-harvest pods. Similarly, (d) displays the updated projection for $\text{ImageSI}_{\text{MDS}^{-1}}$, (e) for $\text{ImageSI}_{\text{DRTriplet}}$, and (f) for $\text{ImageSI}_{\text{PHTriplet}}$.

that by concentrating solely on maturity stages with consistent seed counts (two seeds for all images), we can effectively train the model to understand and capture the concept of pod maturity.

Figure 6.3(a) illustrates the initial projection, while Figure 6.3(b) depicts the user manipulation to categorize the edamame pod images based on the maturity stage. Table 6.3 shows the adjusted Silhouette scores for all frameworks.

Figure 6.3(c) displays the updated projection for WMDS^{-1}. It registers the lowest adjusted Silhouette score among the assessed frameworks. The presence of three red, three blue, and two green contours indicates a failure to adequately segregate clusters based on the maturity stage.

Similarly, Figure 6.3(d) presents the updated projection for $\text{ImageSI}_{\text{MDS}^{-1}}$, which achieves the highest adjusted Silhouette score. However, it misclassifies one diseased pod image, one late-to-harvest pod image, and two ready-to-harvest pod images, more misclassifications than $\text{ImageSI}_{\text{DRTriplet}}$.

In Figure 6.3(e), the updated projection for $\text{ImageSI}_{\text{DRTriplet}}$ is depicted. One misclassified image is evident within the green contour representing ready-to-harvest pods.

Lastly, Figure 6.3(f) illustrates the updated projection for $\text{ImageSI}_{\text{PHTriplet}}$. It shows one misclassified ready-to-harvest pod image and one misclassified diseased pod image.

Table 6.3: Adjusted Silhouette Scores for Ready-to-Harvest vs. Late-to-Harvest vs. Diseased Edamame Pods Dataset

Framework	**Adjusted Silhouette Score**
WMDS^{-1}	0.566
$\text{ImageSI}_{\text{MDS}^{-1}}$	0.880
$\text{ImageSI}_{\text{DRTriplet}}$	0.866
$\text{ImageSI}_{\text{PHTriplet}}$	0.782

6.3.2 Task 2: Number of Seeds

In the second task, our objective is to categorize the edamame pod images based on the number of seeds per pod while also considering the variation in maturity stages.

Figure 6.4(a) illustrates the initial projection, while Figure 6.4(b) depicts the user manipulation to categorize the edamame pod images based on the number of seeds per pod. Table 6.4 shows the adjusted Silhouette scores for all frameworks.

Figure 6.4(c) displays the updated projection for WMDS^{-1}. It registers the lowest adjusted Silhouette score among the frameworks evaluated. Two blue and three green contours indicate a challenge in effectively separating two-seeds and three-seeds edamame pods.

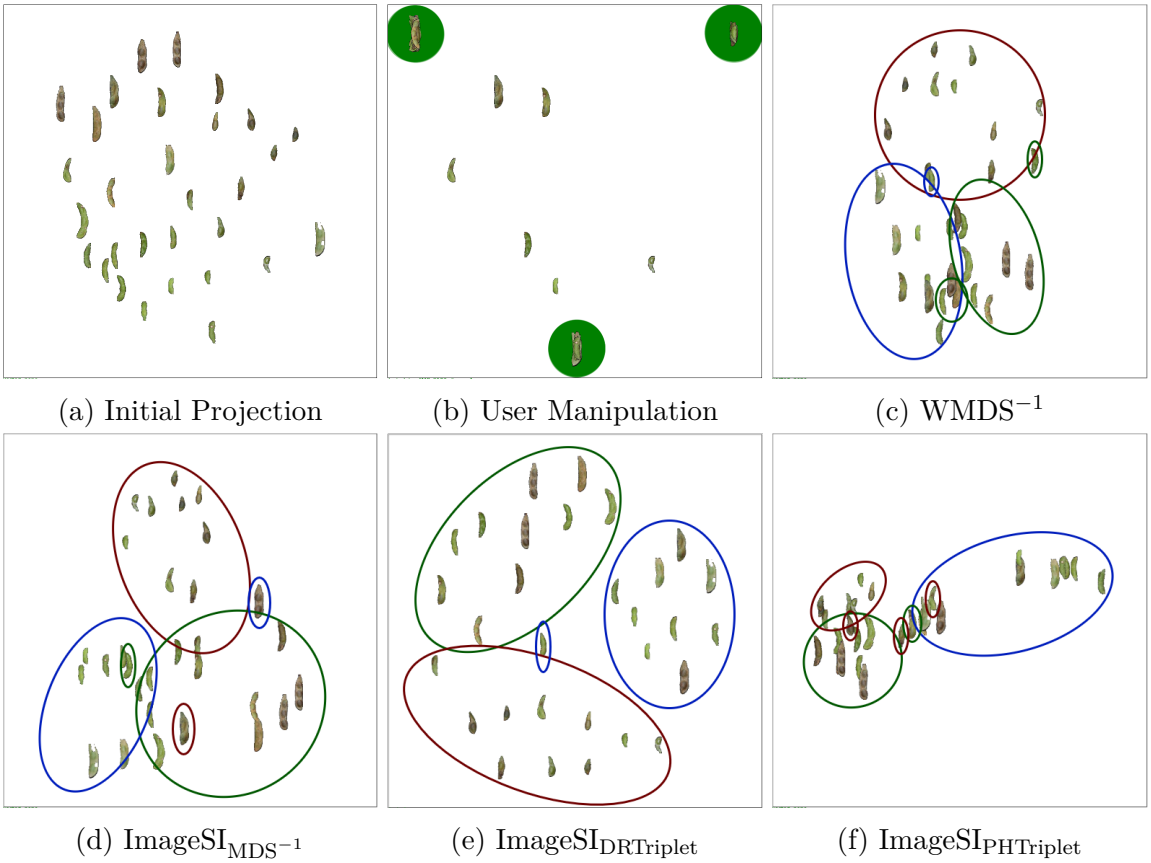
Similarly, Figure 6.4(d) presents the updated projection for $\text{ImageSI}_{\text{MDS}^{-1}}$, revealing two red, two blue, and two green contours, suggesting difficulty in effective cluster separation.

In Figure 6.4(e), the updated projection for $\text{ImageSI}_{\text{DRTriplet}}$ is depicted. One misclassified two-seeds pod is observed.

Lastly, Figure 6.4(f) illustrates the updated projection for $\text{ImageSI}_{\text{PHTriplet}}$. One misclassified image is evident in the green contour representing the three-seeds pod cluster, and three one-seed pods are not assigned to their corresponding cluster. Additionally, some images overlap.

Table 6.4: Adjusted Silhouette Scores for One-seed vs. Two-seeds vs Three-seeds Edamame Pods Dataset

Framework	**Adjusted Silhouette Score**
WMDS^{-1}	0.317
$\text{ImageSI}_{\text{MDS}^{-1}}$	0.554
$\text{ImageSI}_{\text{DRTriplet}}$	0.838
$\text{ImageSI}_{\text{PHTriplet}}$	0.617



(a) Initial Projection (b) User Manipulation (c) WMDS^{-1}

(d) $\text{ImageSI}_{\text{MDS}^{-1}}$ (e) $\text{ImageSI}_{\text{DRTriplet}}$ (f) $\text{ImageSI}_{\text{PHTriplet}}$

Figure 6.4: Usage scenario on the One-seed vs. Two-seeds vs Three-seeds Edamame Pods dataset: (a-f) illustrate the exploratory analysis process on 30 images. In (a), the initial projection is depicted. In (b), the user manipulates the projection by segregating the "one-seed" and "two-seeds" images from the "three-seeds" pod images, emphasizing the concept of "number of seeds" features. The updated projection, (c), showcases the output for WMDS^{-1}, where red contours represent one-seed pods, blue contours represent two-seed pods, and green contours represent three-seed pods. Similarly, (d) displays the updated projection for $\text{ImageSI}_{\text{MDS}^{-1}}$, (e) for $\text{ImageSI}_{\text{DRTriplet}}$, and (f) for $\text{ImageSI}_{\text{PHTriplet}}$.

6.4 Summary

Across the evaluated frameworks, we noticed differences in their ability to accurately separate and cluster images based on their features. $\text{ImageSI}_{\text{DRTriplet}}$ demonstrated clear separation in some cases, such as distinguishing between the two animal datasets. However, it occasionally misclassified edamame pods based on the maturity stage and number of seeds. $\text{ImageSI}_{\text{PHTriplet}}$ and $\text{ImageSI}_{\text{MDS}^{-1}}$ also exhibited promising results but faced challenges at times with dense clustering or overlapping clusters. WMDS^{-1} had the lowest adjusted Silhouette score across all tasks.

Chapter 7

Algorithm-Centered Quantitative Analysis

In our study, we simulated evaluations using augmented datasets from various usage scenarios, each with known ground truth labels. We aimed to quantify and compare the efficiency of each system in achieving satisfactory sorting results based on the number of interactions required. A lower interaction count suggests better efficiency in comprehending and integrating analyst feedback. We calculated and compared adjusted Silhouette scores across all four frameworks.

To evaluate the effectiveness of ImageSI in addressing the tasks outlined in Section 1, we conducted experiments focusing on two primary aspects: first, to compare the inference accuracy of ImageSI frameworks, and second, to evaluate their efficiency in capturing the analyst's intent through interactions.

To quantitatively evaluate the performance of $\text{ImageSI}_{\text{MDS}^{-1}}$, $\text{ImageSI}_{\text{DRTriplet}}$, and $\text{ImageSI}_{\text{PHTriplet}}$, and WMDS^{-1} in organizing image datasets according to human feedback, we employ a simulation-based evaluation approach.

7.1 Experiment Design

Inspired by prior research [5, 12], we designed and developed a simulation engine capable of simulating semantic interactions relevant to individual datasets. These interactions aim to arrange images so that those belonging to the same category are grouped closely together, while those from different categories are distinctly separated.

The simulation process involves conducting simulation iterations until reaching convergence or a predetermined number of iterations. In each iteration, the simulation engine randomly selects samples from each class of the dataset and applies semantic interactions to train the SI systems. It's worth noting that for both the $\text{ImageSI}_{\text{DRTriplet}}$ and $\text{ImageSI}_{\text{PHTriplet}}$ framework, we need at least two samples per category in the dataset to ensure anchor positive pairs; having just one sample per category will not suffice. The systems then learn new DL embeddings based on the feedback provided by the simulation engine, organizing the entire set of images accordingly.

We evaluate the clustering performance using the adjusted Silhouette score, which measures the cohesion and separation of clusters in the projected space. To prevent overfitting, we check for convergence criteria such as early stopping if the loss reaches 0.

7.2 Simulation Engine

To evaluate the performance of the system and assess the effectiveness of semantic interactions, we employ a simulation engine comprising two key components: the Interaction Simulator and the Layout Evaluator.

7.2.1 Interaction Simulator

The interaction simulator simulates semantic interactions to guide the layout of image datasets. For $\text{ImageSI}_{\text{MDS}^{-1}}$ and WMDS^{-1}, the simulator simply selects the specified number of images, k, in each class ("open-mouthed" or "closed-mouthed") and generates a distance matrix such that, for two points x_i and x_j, $||x_i - x_j||$ is 0 if the x_i and x_j are from the same class and $\sqrt{2}$ otherwise.

For $\text{ImageSI}_{\text{Triplet}}$, it selects k samples from each class. It then randomly picks an anchor sample, though each selected sample will be used as an anchor in turn. A positive sample, different from the anchor point, is randomly chosen from the same class as the anchor. Similarly, a negative sample is randomly selected from the points in the other classes. This requires that at least two points per class are moved ($k >= 2$), as selecting only one point would not provide a positive sample for the randomly selected anchor points.

The calculation of the simulated distance can be represented as follows

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

Here, $||x_i - x_j||$ represents the simulated distance between the anchor and positive samples x_i and x_j, respectively. If x_i and x_j belong to the same class, their positions are set to the mean position. Otherwise, $\bar{x}_n$ denotes the mean position of the selected negative samples, representing the position of negative samples relative to the anchor and positive samples.

After simulating the interaction, we apply the corresponding loss function to fine-tune the model. Finally, we extract the updated image embeddings and re-project them using MDS.

7.2.2 Evaluation simulator

After the Interaction Simulator processes the data, the Layout Evaluator assesses the quality of the projected layout. In this evaluation, we utilize the adjusted Silhouette score. We refer to Section 6.1 for detailed information on the use and interpretation of this metric.

7.3 Dataset and Task

In this experiment, we utilize four datasets from the two case studies introduced earlier, each with an expanded data size. The tasks remain consistent with those described in the respective case studies (6.2 and 6.3). We run the simulation engine 10 times for each framework and average the adjusted Silhouette score to obtain a final robust result.

7.4 Result

The comparison of adjusted Silhouette scores across different frameworks and tasks is illustrated in Figure 7.1. Subfigures (a) to (d) depict the performance of $\text{ImageSI}_{\text{DRTriplet}}$, $\text{ImageSI}_{\text{PHTriplet}}$, WMDS^{-1}, and $\text{ImageSI}_{\text{MDS}^{-1}}$, respectively, in various sorting tasks. Each subplot shows the adjusted Silhouette scores achieved by the frameworks over a range of interaction counts.

In the task comparing Open-Mouth vs. Closed-Mouth Animals (Figure 7.1a), $\text{ImageSI}_{\text{DRTriplet}}$ and $\text{ImageSI}_{\text{MDS}^{-1}}$ consistently outperformed the other frameworks across all interaction counts. Specifically, $\text{ImageSI}_{\text{DRTriplet}}$ achieved adjusted Silhouette scores ranging from 0.008 to 0.544, while $\text{ImageSI}_{\text{MDS}^{-1}}$ exhibited scores within the range of 0.008 to 0.599. $\text{ImageSI}_{\text{PHTriplet}}$ also showcased competitive performance, particularly at higher interaction counts. However,

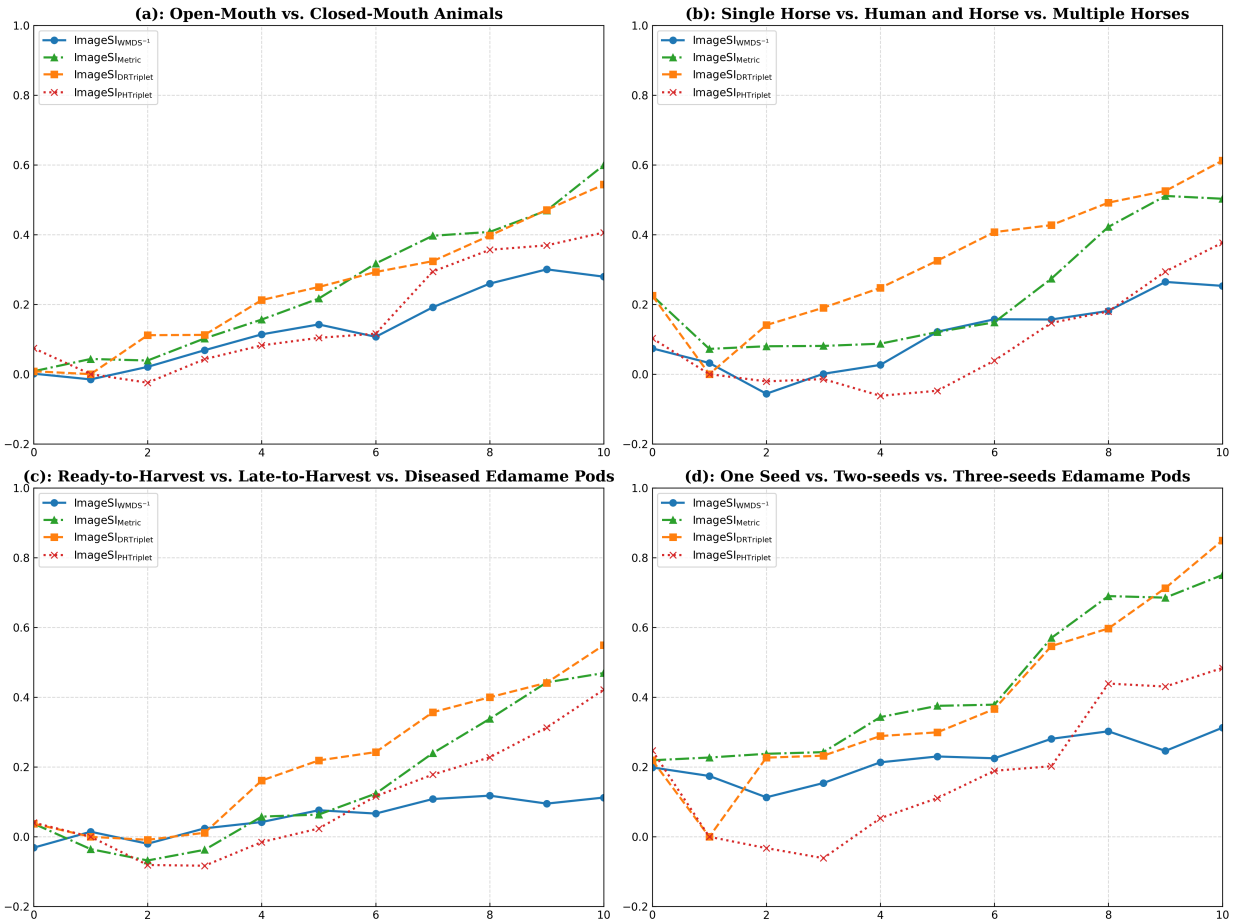


Figure 7.1: Comparison of adjusted Silhouette scores across different frameworks and tasks. Subfigures (a) to (d) depict the performance of WMDS^{-1}, and ImageSI$_{\text{MDS}^{-1}}$, ImageSI$_{\text{DRTriplet}}$, and ImageSI$_{\text{PHTriplet}}$, respectively, in various sorting tasks. Each subplot shows the adjusted Silhouette scores achieved by the frameworks over a range of interaction counts.

WMDS^{-1} exhibited comparatively lower performance throughout the range of interaction counts.

For the sorting task comparing Single Horse vs. Human and Horse vs. Multiple Horses (Figure 7.1b), $\text{ImageSI}_{\text{DRTriplet}}$ exhibited notable performance across all interaction counts, achieving silhouette scores ranging from 0.225 to 0.612. $\text{ImageSI}_{\text{MDS}^{-1}}$ also demonstrated competitive performance, especially at higher interaction counts, with scores ranging from 0.225 to 0.503. Conversely, WMDS^{-1} and $\text{ImageSI}_{\text{PHTriplet}}$ displayed comparatively lower performance, with adjusted Silhouette scores varying from 0.074 to 0.253 and 0.103 to 0.377, respectively.

In the task comparing Ready-to-Harvest vs. Late-to-Harvest vs. Diseased Edamame Pods (Figure 7.1c), Both $\text{ImageSI}_{\text{MDS}^{-1}}$ and $\text{ImageSI}_{\text{DRTriplet}}$ consistently demonstrated superior performance across the interaction counts, particularly excelling at higher interaction counts. Achieving adjusted Silhouette scores ranged from 0.219 to 0.751 and 0.219 to 0.850, respectively. $\text{ImageSI}_{\text{PHTriplet}}$ presented a competitive performance in this task, with an adjusted Silhouette score ranging from 0.247 to 0.484. In contrast, WMDS^{-1} showcased comparatively lower performance, with silhouette scores ranging from 0.199 to 0.313.

Lastly, in the task comparing One-seed vs. Two-seeds vs. Three-seeds Edamame Pods (Figure 7.1d), $\text{ImageSI}_{\text{DRTriplet}}$ consistently showcased the highest performance throughout most interaction counts, exhibiting silhouette scores ranging from 0.038 to 0.550. $\text{ImageSI}_{\text{MDS}^{-1}}$ demonstrated competitive performance, particularly evident at higher interaction counts. Its silhouette scores ranged from 0.038 to 0.469. In contrast, WMDS^{-1} displayed relatively lower performance across all interaction counts, with silhouette scores ranging from -0.031 to 0.113. Similarly, $\text{ImageSI}_{\text{PHTriplet}}$ exhibited performance inferior to $\text{ImageSI}_{\text{DRTriplet}}$ and $\text{ImageSI}_{\text{MDS}^{-1}}$, with silhouette scores ranging from 0.041 to 0.422.

Overall, $\text{ImageSI}_{\text{DRTriplet}}$ and $\text{ImageSI}_{\text{MDS}^{-1}}$ consistently demonstrated superior performance in terms of both inference accuracy and interaction efficiency, as measured by the adjusted Silhouette score. Following closely was $\text{ImageSI}_{\text{PHTriplet}}$. Conversely, WMDS^{-1} generally exhibited lower performance levels.

Chapter 8

Discussion

8.1 Loss Functions

In interactive deep metric learning, traditional metric loss like triplet loss [39], contrastive loss [31], angular loss [44], quadruplet loss [45], N-Pair loss [46], and Histogram loss [47] have been used to shape the representation learned by the model. In this work, we employ triplet loss and MDS^{-1} to guide the DL model in capturing user intention. Triplet loss optimizes the embedding space based on relative distances between samples. However, it largely disregards the actual pairwise distances between data points, only using them to infer clusters of images, which potentially overlooks meaningful feedback. In contrast, MDS^{-1} aligns pairwise distances in the embedding space with those in the DR space but does not create as cleanly organized groups of images. Integrating MDS^{-1} with triplet loss would address this limitation by incorporating pairwise distances into the learning process, while still emphasizing clustering. This would pair the detailed feedback from MDS^{-1} with the cluster's superior organizational abilities of the triplet. The integration involves using recovered pairwise distances to guide learning, enhancing the model's ability to effectively capture local and global structures.

8.2 Balancing Inference Accuracy and Cluster Dispersion

In evaluating the performance of $\text{ImageSI}_{\text{MDS}^{-1}}$ and $\text{ImageSI}_{\text{PHTriplet}}$, we observed competitive or superior adjusted Silhouette scores across both case studies and algorithm-centered quantitative analyses. Notably, $\text{ImageSI}_{\text{MDS}^{-1}}$ and $\text{ImageSI}_{\text{PHTriplet}}$ outperformed $\text{ImageSI}_{\text{DRTriplet}}$ in two out of four case studies.

However, despite the promising results in adjusted Silhouette scores, both $\text{ImageSI}_{\text{MDS}^{-1}}$ and $\text{ImageSI}_{\text{PHTriplet}}$ face challenges with cluster overlapping. While these models effectively minimize distances within clusters, they occasionally produce clusters that are too dense and crowded. While cohesive clusters are desirable for clear separation, overly dense clusters may obscure valuable insights inherent in cluster dispersion.

To address this trade-off between inference accuracy and clustering dispersion, one potential solution is to introduce a regularization term in the model pipeline that penalizes excessive cluster density. For example, the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm incorporates a parameter called "minPts," which determines the minimum number of points required to form a dense region or cluster [48]. By adjusting this parameter, users can control the density of clusters, effectively penalizing overly dense clusters and promoting more balanced dispersion [49]. However, choosing the appropriate parameters is challenging, particularly when there is significant density variation within clusters. [50] introduced a method for determining optimal parameter values for different types of clusters by detecting sharp distance increases through a function calculating distances between each dataset element and its k-th nearest neighbor.

Ester et al. introduced the DBSCAN algorithm and demonstrated its effectiveness in clus-

tering spatial data with varying densities [51]. The algorithm's ability to adapt to different density levels makes it suitable for addressing the trade-off between accuracy and clustering dispersion.

Additionally, exploring ensemble clustering techniques or incorporating DR methods, such as t-SNE, could offer alternative approaches to mitigating cluster overlap while preserving cluster separation. These techniques have been widely studied and applied in various domains to enhance clustering performance and interoperability [52, 53, 54].

Finding the right balance between accurately predicting outcomes and grouping similar data points is key for making sure the information we gather from analyzing data is both valuable and useful. By tackling issues like overlapping clusters head-on and using the methods mentioned earlier, we can boost the efficiency of tools that help users explore data interactively.

8.3 Interactive Deep Learning Model

The ImageSI framework uses a fine-tuned pre-trained ResNet-18 model as a feature extractor to enable data projection. Through user interaction, data points are dynamically repositioned within the projection, refining the model's understanding iteratively. This process involves model backpropagation, where the ResNet model with updated convolutional layers adjusts to capture the analyst's intent better, ensuring continual alignment with user expectations and data characteristics.

While ResNet-18 forms the basis for feature extraction, exploring alternative deep-learning models holds the potential for further enhancing the framework's capabilities. One viable option is to consider deeper variants within the ResNet family, such as ResNet-50 or ResNet-101. These models offer increased depth, enabling them to capture more intricate feature

representations and potentially discern finer nuances within the data. For instance, ResNet-50 has demonstrated superior performance in image classification tasks compared to ResNet-18, attributable to its deeper architecture [55].

Another promising avenue involves the adoption of Vision Transformers (ViTs). Unlike traditional convolutional neural networks (CNNs) like ResNet, ViTs rely solely on self-attention mechanisms, making them adept at capturing global dependencies within the data. Recent studies have showcased the efficacy of ViTs across various computer vision tasks, often outperforming CNN-based models [56, 57, 58]. Integrating ViTs into the ImageSI framework could enhance the model's ability to capture intricate relationships and semantic structures within the data projection.

Moreover, attention-based models like Vision Transformers provide interpretability advantages, allowing analysts to understand how different input components contribute to final predictions, enriching the exploration process [56].

The ImageSI framework is very flexible and can easily be integrated with other DL models for feature extraction. This offers exciting opportunities to improve interactive data exploration and analysis. By using advanced models like ResNet variants and new architectures like Vision Transformers, the framework can better adapt to user needs, allowing analysts to gain more profound insights from their data.

8.4 Dimensionality Reduction Techniques

In our framework, we utilize MDS as a nonlinear DRT. MDS facilitates the visualization of high-dimensional data by projecting it into a lower-dimensional space while preserving the pairwise distances between data points [59]. Despite its simplicity and interpretability, MDS

has limitations that may constrain its applicability [60].

As a nonlinear DRT, MDS faces limitations when applied to complex relationships inherent in real-world datasets [61]. This can lead to suboptimal embeddings, where the lower-dimensional representation fails to accurately capture the true underlying relationships among data points. Furthermore, the curse of dimensionality, which states that pairwise distances become increasingly sparse and less significant as data dimensionality rises, makes MDS less effective with high-dimensional data [62].

To address these limitations, more advanced nonlinear DRTs offer improved capabilities for representing complex data structures. Techniques such as autoencoders, t-SNE, and UMAP have demonstrated effectiveness in capturing nonlinear relationships and preserving the intrinsic structure of high-dimensional data [30, 63, 64].

Autoencoders, as neural network-based models, are particularly adept at learning nonlinear mappings from high-dimensional data to lower-dimensional representations [65]. They offer flexibility and adaptability in capturing intricate data patterns, making them suitable for a wide range of datasets.

Similarly, t-SNE and UMAP have gained attention for their ability to preserve both local and global data structures in high-dimensional spaces [63, 64]. t-SNE excels in producing high-quality visualizations by using random walks on neighborhood graphs to overcome crowding issues, while UMAP creates topological representations of data to maintain local and global relationships effectively.

By using advanced techniques like Autoencoders, t-SNE, and UMAP, our system can produce more useful embeddings that represent the data structure well and effectively handle complex high-dimensional data.

8.5 Visual Explanation

In the baseline model, WMDS^{-1}, Han et al. provide explanations of learned features through weighted saliency maps [12]. Their visual explanations rely on Weighted Visual Backpropagation, utilizing inverse weighted projections from the Interactive DR loop to guide user feedback through the neural network's convolutional layers. However, in the $\text{ImageSI}_{\text{MDS}^{-1}}$, $\text{ImageSI}_{\text{DRTriplet}}$ and $\text{ImageSI}_{\text{PHTriplet}}$ frameworks, the use of WMDS^{-1} is not employed. As a result, weights optimized for preserving specified relationships cannot be computed. Without these weights, Weighted Visual Backpropagation cannot function effectively, posing challenges in providing meaningful visual explanations for model predictions.

One solution to overcome this limitation is to utilize the Visual Backpropagation method [66]. This approach effectively calculates neuron contributions to feature representation, allowing fast backpropagation. Although lacking the projection-specific optimization of WMDS^{-1}, implementing the Visual Backpropagation method enables the framework to continue offering visual insights into the model's decision-making process.

We can continue to improve the interpretability of the model's decision by developing visual explanation techniques that are appropriate for the limitations of the framework. This will enable users to obtain a deeper understanding of the underlying mechanisms that underlie the model's predictions.

Chapter 9

Conclusion

In this work, we introduced the ImageSI framework to facilitate interactive DL for image data analysis within the context of VA. Utilizing SI techniques, ImageSI refines data projections iteratively based on user feedback, enabling users to explore and understand complex image datasets interactively. Unlike traditional dimensionality reduction methods, our framework leverages DL models for interactive sensemaking without needing labeled data or specific features. We evaluated three variants of the ImageSI framework $\text{ImageSI}_{\text{MDS}^{-1}}$, $\text{ImageSI}_{\text{DRTriplet}}$, and $\text{ImageSI}_{\text{PHTriplet}}$, against the baseline model, WMDS^{-1}, through various usage scenarios and algorithm-centered quantitative analyses. Our findings indicate that $\text{ImageSI}_{\text{DRTriplet}}$ and $\text{ImageSI}_{\text{MDS}^{-1}}$ consistently outperform other frameworks in terms of inference accuracy and interaction efficiency, achieving higher adjusted Silhouette scores across different image projection tasks. Moreover, $\text{ImageSI}_{\text{PHTriplet}}$ demonstrated competitive performance, while WMDS^{-1} generally exhibited lower performance metrics.

Bibliography

[1] J. Thomas, *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE, 2005.

[2] A. Endert, R. Chang, C. North, and M. Zhou, "Semantic interaction: Coupling cognition and computation through usable interactive analytics," *IEEE Computer Graphics and Applications*, vol. 35, no. 4, pp. 94–99, 2015.

[3] A. Endert, P. Fiaux, and C. North, "Semantic interaction for visual text analytics," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, (New York, NY, USA), p. 473–482, Association for Computing Machinery, 2012.

[4] J. Z. Self, L. House, S. Leman, and C. North, "Andromeda: Observation-level and parametric interaction for exploratory data analysis," *Technical report, Technical report*, 2015.

[5] Y. Bian and C. North, "Deepsi: Interactive deep learning for semantic interaction," in *26th International Conference on Intelligent User Interfaces*, pp. 197–207, 2021.

[6] D. Sacha, L. Zhang, M. Sedlmair, J. A. Lee, J. Peltonen, D. Weiskopf, S. C. North, and D. A. Keim, "Visual interaction with dimensionality reduction: A structured literature analysis," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 1, pp. 241–250, 2016.

[7] J. Z. Self, R. K. Vinayagam, J. T. Fry, and C. North, "Bridging the gap between user intention and model parameters for human-in-the-loop data analytics," in *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*, pp. 1–6, 2016.

[8] H. K. Palo, S. Sahoo, and A. K. Subudhi, "Dimensionality reduction techniques: Principles, benefits, and limitations," *Data Analytics in Bioinformatics: A Machine Learning Perspective*, pp. 77–107, 2021.

[9] Y. Bian, M. Dowling, and C. North, "Evaluating semantic interaction on word embeddings via simulation," *arXiv preprint arXiv:2007.15824*, 2020.

[10] S. Gehrmann, H. Strobelt, R. Krüger, H. Pfister, and A. M. Rush, "Visual interaction with deep learning models through collaborative semantic inference," *IEEE transactions on visualization and computer graphics*, vol. 26, no. 1, pp. 884–894, 2019.

[11] B. C. Kwon, M.-J. Choi, J. T. Kim, E. Choi, Y. B. Kim, S. Kwon, J. Sun, and J. Choo, "Retainvis: Visual analytics with interpretable and interactive recurrent neural networks on electronic medical records," *IEEE transactions on visualization and computer graphics*, vol. 25, no. 1, pp. 299–309, 2018.

[12] H. Han, R. Faust, B. F. Keith Norambuena, J. Lin, S. Li, and C. North, "Explainable interactive projections of images," *Machine Vision and Applications*, vol. 34, no. 6, p. 100, 2023.

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

[14] S. Banerjee, "Animal image dataset." https://www.kaggle.com/datasets/iamsouravbanerjee/animal-image-dataset-90-different-animals?select=animals.

[15] H. Han, R. Prabhu, T. Smith, K. Dhakal, X. Wei, S. Li, and C. North, "Interactive deep learning for exploratory sorting of plantimages by visual phenotypes," 2022.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[17] P. Achenbach, D. Purdack, S. Wolf, P. N. Müller, T. Tregel, and S. Göbel, "Paper beats rock: Elaborating the best machine learning classifier for hand gesture recognition," in *Joint International Conference on Serious Games*, pp. 229–245, Springer, 2022.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pp. 630–645, Springer, 2016.

[19] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

[20] A. Boyd, A. Czajka, and K. Bowyer, "Deep learning-based feature extraction in iris recognition: Use existing models, fine-tune or train from scratch?," in *2019 IEEE 10th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pp. 1–9, IEEE, 2019.

[21] J. Wenskovitch, I. Crandell, N. Ramakrishnan, L. House, and C. North, "Towards a systematic combination of dimension reduction and clustering in visual analytics," *IEEE transactions on visualization and computer graphics*, vol. 24, no. 1, pp. 131–141, 2017.

[22] J. Z. Self, X. Hu, L. House, S. Leman, and C. North, "Designing usable interactive visual analytics tools for dimension reduction," in *CHI 2016 Workshop on Human-Centered Machine Learning (HCML)*, p. 7, 2016.

[23] R. Zebari, A. Abdulazeez, D. Zeebaree, D. Zebari, and J. Saeed, "A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction," *Journal of Applied Science and Technology Trends*, vol. 1, no. 1, pp. 56–70, 2020.

[24] S. Ayesha, M. K. Hanif, and R. Talib, "Overview and comparative study of dimensionality reduction techniques for high dimensional data," *Information Fusion*, vol. 59, pp. 44–58, 2020.

[25] A. Maćkiewicz and W. Ratajczak, "Principal components analysis (pca)," *Computers & Geosciences*, vol. 19, no. 3, pp. 303–342, 1993.

[26] R. Abdulhammed, H. Musafer, A. Alessa, M. Faezipour, and A. Abuzneid, "Features dimensionality reduction approaches for machine learning based network intrusion detection," *Electronics*, vol. 8, no. 3, 2019.

[27] Y. Wang, H. Huang, C. Rudin, and Y. Shaposhnik, "Understanding how dimension reduction tools work: an empirical approach to deciphering t-sne, umap, trimap, and pacmap for data visualization," *Journal of Machine Learning Research*, vol. 22, no. 201, pp. 1–73, 2021.

[28] K. Pal and M. Sharma, "Performance evaluation of non-linear techniques umap and t-sne for data in higher dimensional topological space," in *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, pp. 1106–1110, 2020.

[29] M. Espadoto, N. S. T. Hirata, and A. C. Telea, "Deep learning multidimensional projections," 2019.

[30] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[31] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, vol. 2, pp. 1735–1742, IEEE, 2006.

[32] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2022.

[33] A. González Martínez, B. T. Wooton, N. Kirshenbaum, D. Kobayashi, and J. Leigh, "Exploring collections of research publications with human steerable ai," in *Practice and Experience in Advanced Research Computing*, pp. 339–348, 2020.

[34] A. Endert, P. Fiaux, and C. North, "Semantic interaction for visual text analytics," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 473–482, 2012.

[35] D. Sacha, A. Stoffel, F. Stoffel, B. C. Kwon, G. Ellis, and D. A. Keim, "Knowledge generation model for visual analytics," *IEEE transactions on visualization and computer graphics*, vol. 20, no. 12, pp. 1604–1613, 2014.

[36] J. S. Yi, Y. a. Kang, J. Stasko, and J. Jacko, "Toward a deeper understanding of the role of interaction in information visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1224–1231, 2007.

[37] X. Chen, J. Z. Self, L. House, and C. North, "Be the data: A new approach for lmmersive analytics," in *2016 Workshop on Immersive Analytics (IA)*, pp. 32–37, IEEE, 2016.

[38] M. Wang, J. Wenskovitch, L. House, N. Polys, and C. North, "Bridging cognitive gaps between user and model in interactive dimension reduction," *Visual Informatics*, vol. 5, no. 2, pp. 13–25, 2021.

[39] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face

recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[40] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," 2014.

[41] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," 2013.

[42] S. Dara and P. Tumma, "Feature extraction by using deep learning: A survey," in *2018 Second international conference on electronics, communication and aerospace technology (ICECA)*, pp. 1795–1801, IEEE, 2018.

[43] S. Lefranc, P. Roca, M. Perrot, C. Poupon, D. Le Bihan, J.-F. Mangin, and D. Rivière, "Groupwise connectivity-based parcellation of the whole human cortical surface using watershed-driven dimension reduction," *Medical Image Analysis*, vol. 30, pp. 11–29, 2016.

[44] J. Wang, F. Zhou, S. Wen, X. Liu, and Y. Lin, "Deep metric learning with angular loss," in *Proceedings of the IEEE international conference on computer vision*, pp. 2593–2601, 2017.

[45] W. Chen, X. Chen, J. Zhang, and K. Huang, "Beyond triplet loss: a deep quadruplet network for person re-identification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 403–412, 2017.

[46] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," *Advances in neural information processing systems*, vol. 29, 2016.

[47] E. Ustinova and V. Lempitsky, "Learning deep embeddings with histogram loss," *Advances in neural information processing systems*, vol. 29, 2016.

[48] M. Kotyrba, E. Volná, and Z. Komínková Oplatková, "Comparison of modern clustering algorithms for twodimensional data," in *Proceedings-28th European Conference on Modelling and Simulation, ECMS 2014*, European Council for Modelling and Simulation, 2014.

[49] L. Ma, "An improved and heuristic-based iterative dbscan clustering algorithm," in *2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, vol. 5, pp. 2709–2714, IEEE, 2021.

[50] A. Starczewski, P. Goetzen, and M. J. Er, "A new method for automatic determining of the dbscan parameters," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 10, no. 3, pp. 209–221, 2020.

[51] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *kdd*, vol. 96, pp. 226–231, 1996.

[52] S. Arora, W. Hu, and P. K. Kothari, "An analysis of the t-sne algorithm for data visualization," in *Proceedings of the 31st Conference On Learning Theory* (S. Bubeck, V. Perchet, and P. Rigollet, eds.), vol. 75 of *Proceedings of Machine Learning Research*, pp. 1455–1462, PMLR, 06–09 Jul 2018.

[53] H. Liu, J. Yang, M. Ye, S. C. James, Z. Tang, J. Dong, and T. Xing, "Using t-distributed stochastic neighbor embedding (t-sne) for cluster analysis and spatial zone delineation of groundwater geochemistry data," *Journal of Hydrology*, vol. 597, p. 126146, 2021.

[54] G. C. Linderman and S. Steinerberger, "Clustering with t-sne, provably," *SIAM journal on mathematics of data science*, vol. 1, no. 2, pp. 313–332, 2019.

[55] Z. Wu, C. Shen, and A. Van Den Hengel, "Wider or deeper: Revisiting the resnet model for visual recognition," *Pattern recognition*, vol. 90, pp. 119–133, 2019.

[56] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[57] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*, pp. 213–229, Springer, 2020.

[58] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *International conference on machine learning*, pp. 10347–10357, PMLR, 2021.

[59] M. A. A. Cox and T. F. Cox, *Multidimensional Scaling*, pp. 315–347. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[60] I. Borg and P. J. Groenen, *Modern multidimensional scaling: Theory and applications*. Springer Science & Business Media, 2005.

[61] N. Saeed, H. Nam, M. I. U. Haq, and D. B. Muhammad Saqib, "A survey on multidimensional scaling," *ACM Computing Surveys (CSUR)*, vol. 51, no. 3, pp. 1–25, 2018.

[62] M. Köppen, "The curse of dimensionality," in *5th online world conference on soft computing in industrial applications (WSC5)*, vol. 1, pp. 4–8, 2000.

[63] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne.," *Journal of machine learning research*, vol. 9, no. 11, 2008.

[64] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.

[65] P. Baldi, "Autoencoders, unsupervised learning, and deep architectures," in *Proceedings of ICML Workshop on Unsupervised and Transfer Learning* (I. Guyon, G. Dror, V. Lemaire, G. Taylor, and D. Silver, eds.), vol. 27 of *Proceedings of Machine Learning Research*, (Bellevue, Washington, USA), pp. 37–49, PMLR, 02 Jul 2012.

[66] M. Bojarski, A. Choromanska, K. Choromanski, B. Firner, L. Jackel, U. Muller, and K. Zieba, "Visualbackprop: Efficient visualization of cnns," *arXiv preprint arXiv:1611.05418*, 2016.