Data Mining and Knowledge Discovery (2025) 39:73
https://doi.org/10.1007/s10618-025-01154-y

Stable graph based decision route explanation in siamese neural networks

Rabia Saleem[1] · **Ashiq Anjum**[1] · **Bo Yuan**[1] · **Lu Liu**[1]

Received: 2 December 2023 / Accepted: 30 July 2025 / Published online: 19 August 2025
© The Author(s) 2025

Abstract

Siamese Neural Networks (SNNs) have shown promise in addressing a variety of tasks, even with limited data availability. However, their adoption is hindered by the lack of transparency in their decision-making processes. A key challenge in explaining SNNs lies in the absence of an inverse mapping between high-dimensional input feature vectors and the low-dimensional embedding space. Therefore, computing direct distances between input features becomes meaningless. Existing autoencoder-based explanation methods face several limitations. These include poor image reconstruction quality due to insufficient data and the omission of final distance layer of the SNN during the explanation process. While the Siamese Network Explainer (SINEX) can explain audio and grayscale images, it does not support RGB images. To overcome these challenges, we propose a method called Features Distance-based eXplanation (FDbX). This approach identifies salient features using ridge regression, trained on perturbed SLIC-segmented images. To enhance the selection of important features, we incorporate Bayesian analysis, which assigns importance scores to features. To provide a comprehensive explanation of the decision route, we construct a mathematical model that represents important features and their Hamming distances as a bipartite graph. In this graph, nodes represent features and edges denote distances between feature pairs. The resulting explanation heatmaps highlight critical image segments, offering more intuitive and visually informative explanations than existing methods. We evaluate stability and faithfulness of our method using stability indices such as R^2 and mean squared error. To the best of our knowledge, this is the first work to introduce Variable and Coefficient Stability Indices for image datasets.

Keywords SNNs · Explainable AI · Hamming distance · Bipartite graph

Extended author information available on the last page of the article

1 Introduction

Deep Neural Networks (DNNs) have gained significant popularity due to their superior performance in terms of both accuracy and stability. These models enable machine learning (ML) systems to deliver reliable predictions, even for highly complex tasks such as medical diagnosis and autonomous driving. DNNs often rely on comparing objects in pairs—such as data vectors or image pairs—to learn meaningful relationships. Distance metric learning is a key approach used to address this challenge by computing pairwise metrics that quantify the similarity or dissimilarity between data vectors. The core idea behind metric learning is that dissimilar objects should be positioned farther apart in the feature space than similar ones (Leventi-Peetz and Östreich 2022). Siamese Neural Networks (SNNs) are among the most fundamental architectures for implementing metric learning. They consist of two or more identical subnetworks and use distance-based learning techniques to distinguish between previously unseen classes, categorizing them into semantically similar or dissimilar instances (Roy et al. 2019).

SNNs (Bromley et al. 1993a) were first introduced by Bromley and LeCun to address image matching problems, such as signature verification. In this setup, the twin network receives two distinct signature images and computes a similarity score between them. If the signatures are identical, the network outputs a "Yes"; otherwise, it returns a "No" (Bromley et al. 1993b). Extensive research has also been conducted on image matching algorithms like Oriented FAST and Rotated BRIEF (Zhao et al. 2022), which extract features to generate descriptors. These descriptors are then compared using Hamming distance to determine image similarity. One of the key advantages of SNNs is their robustness to class imbalance, meaning they can perform well even with only a few samples per class—making them particularly effective for matching and verification tasks. However, despite their strong performance in classification and matching, SNNs remain largely opaque in terms of interpretability. Explaining the predictions or decisions made by SNNs is essential for building trust and reliability, especially in safety-critical domains such as medical diagnosis (Wilming et al. 2022).

Explainable Artificial Intelligence (XAI) [7] is an emerging field focused on making machine learning models more transparent by providing insights into how predictions are made, typically through textual or visual explanations. However, popular explainability techniques such as Local Interpretable Model-Agnostic Explanations (LIME) (Garreau and Mardaoui 2021) are not suitable for Siamese Neural Networks (SNNs) due to their unique input structure. SNNs operate on image pairs, whereas LIME is designed for single-instance inputs. Although several variants of LIME exist, none are specifically designed to handle models that require paired inputs. Explaining SNNs presents several challenges. First, the input instances are either similar or dissimilar pairs, making direct comparisons between individual features ineffective. Second, there is no inverse mapping from the low-dimensional embedding space back to the original high-dimensional input space, which complicates interpretation. Finally, due to the high dimensionality of the input data, common perturbation techniques—such as sliding gray squares, superpixel occlusion, or random pattern masking—cannot be directly applied to paired inputs (Utkin et al. 2021). These limitations

highlight the need for specialized explainability methods tailored to the architecture and input format of SNNs.

In addition to the challenges mentioned above, existing explanation methods for SNNs have several other limitations. For instance, the method proposed in Sheng and Huber (2019) introduces a gradient-based approach that involves training a separate autoencoder (AE) to generate explanations. However, this method does not incorporate the final distance layer of the SNN and instead focuses only on the embedding layer—a hidden layer that maps high-dimensional input data (such as images) into a lower-dimensional vector space. Due to limited data availability, the pre-trained decoder in this approach struggles to reconstruct high-quality images, which compromises the overall quality of the explanation. Similarly, the method proposed in Fedele et al. (2022) claims to effectively explain SNN decisions for audio datasets and grayscale images, but it is not suitable for RGB images. Furthermore, none of the aforementioned studies provide evaluation metrics to assess the quality or reliability of their explanations, leaving a gap in the validation of these methods.

In this paper, we propose a post-hoc, perturbation-based method that provides an explanation of the decision-making process from the input layer to the final distance layer of a SNN. Each image in a pair, along with its corresponding subnetwork, is treated as an independent model. To explain the behavior of each subnetwork, we train a regression model using randomly generated samples obtained by perturbing a SLIC-segmented version of the image. Simple Linear Iterative Clustering (SLIC) (Schallner et al. 2020) is a widely used algorithm that segments an image into superpixels, making it easier to analyze and interpret. To enhance the selection of meaningful features, we apply Bayesian statistical analysis, which assigns importance scores to features and filters out those that are not statistically significant. This process is repeated for the second image in the pair, and the corresponding subnetwork highlights the features that contribute to its decision. The final output is a heatmap that visualizes the most influential features, offering a clear and interpretable explanation of the model's decision. The next step in our approach involves calculating the Hamming distance between the important features identified in the previous stage. Using this information, we construct an undirected raw graph, denoted as G_s, where the nodes represent superpixels or features, and the edges represent the Hamming distances between corresponding features. This graph is structured as a bipartite graph (Rong et al. 2022; Caro-Martínez et al. 2023) meaning its nodes can be divided into two disjoint sets, with all edges connecting nodes from one set to nodes in the other. To enhance interpretability, we apply a similarity threshold of 50%, removing all edges whose Hamming distance exceeds this threshold. This filtering process transforms the dense bipartite graph into a sparse bipartite graph, which serves as a visual and structural explanation of the SNN's decision-making process.

We evaluate the quality of the generated explanations using local fidelity, which measures how well our proposed method replicates the behavior of the SNN around specific predictions. Additionally, we assess the stability of the explanations (Visani et al. 2022) to ensure consistency despite minor variations in the data or model. To this end, we introduce two metrics: the Variable Stability Index (VSI) and the Coefficient Stability Index (CSI). These indices evaluate the robustness of the explanation by analyzing the consistency of selected features and their associated weights across

different runs. To the best of our knowledge, this is the first time that VSI and CSI have been applied to assess the stability of explanation methods for image datasets.

To summarize, this paper introduces a novel method called Features Distance-based eXplanation (FDbX) for explaining why a given pair of images is classified as belonging to the same class by a Siamese Neural Network (SNN). The key contributions of this work are as follows:

1. We propose the FDbX method to explain the decision-making process from the input layer to the final distance layer of an SNN, applicable to both grayscale and RGB images. The method employs model distillation to identify and correlate the most influential features contributing to the prediction.
2. We construct an explanation bipartite graph, which serves as a deterministic model to visualize and interpret the SNN's decision pathway.
3. To the best of our knowledge, this is the first study to introduce and apply explanation stability indices—namely, the Variable Stability Index (VSI) and Coefficient Stability Index (CSI)—for evaluating explanation consistency on image datasets.
4. We present both visual and stability-based comparisons, demonstrating that the proposed FDbX method offers more intuitive, interpretable, and reliable explanations than existing SNN explanation techniques.

The remainder of this paper is organized as follows: Sect. 2 reviews the state-of-the-art methods for explaining SNN decisions. Section 3 defines the research problem. Section 4 presents the proposed Features Distance-based eXplanation (FDbX) method, which explains the decision-making process of SNNs. Section 5 describes the experimental setup, while Sect. 6 discusses the key findings. Comparison of the proposed method with existing approaches is discussed in Sect. 7. Finally, Sect. 8 concludes the paper and outlines directions for future research.

2 Related work

There are numerous attempts to explain the rationale behind DNN's black-box decisions. However, some approaches explain one particular decision (local explanation) while other methods provide a complete roadmap from the architecture to all possible decisions (global explanation). The XAI methods can be categorised by scope (local and global), approach (backpropagation and perturbation), and usage (ante-hoc and post-hoc) (Saleem et al. 2022). The few prominent explanation methods such as LIME (Garreau and Mardaoui 2021), SHapley Additive exPlanations (SHAP) (Antwarg et al. 2021), and Gradient-weighted Class Activation Mapping (Grad-CAM) (Chakraborty et al. 2022) have been used to explain DNN models such as Convolutional Neural Networks (CNNs) and Residual Neural Networks (ResNets) (Riaz and Smeaton 2023).

As an agnostic approach, LIME identifies the regions of an image that are most likely to influence the decisions of a predictive model (Ribeiro et al. 2016). In order to explain the prediction, LIME algorithm decomposes an image into superpixels

using quickshift segmentation algorithm based on color or brightness similarities. The random perturbation of segmented images generates new samples to train the surrogate model for each instance. The coefficients of the trained surrogate model help to highlight the important superpixels of the image. Over the past few years, there have been a lot of modifications in the LIME algorithm such as D-LIME (Zafar and Khan 2019) that is a deterministic way to explain the prediction by using Agglomerative Hierarchical Clustering (AHC) to group the training data, instead of random sampling. Alpeh-LIME (Rabold et al. 2020) divides the image into irregular segments and uses the core concept of Alpeh to extract symbolic rules from images. These rules then explain the prediction of a classifier. Next, a hybrid approach, Guided-LIME (Sangroya et al. 2020) uses formal concept analysis to structure the training data then LIME algorithm explains the outcome predicted with this structured data. M-LIME (Shi et al. 2020) modified sampling technique of LIME converts superpixels into undirected graphs. A clique set construction problem is formalized to generate samples and to find the correlation between features whereas Q-LIME (Bramhall et al. 2020), a quadratic version, extends the flexibility of LIME to explain the non-linear or quadratic relationships. A discussion of improving LIME's fidelity is presented in Smooth (s-LIME) (Gaudel et al. 2022). These versions are used to explain different architectures such as random forest, Google's pre-trained Inception, CNNs, MobileNet, XGBoost, and ResNet. Although LIME has a reasonable degree of accuracy and is widely used for explaining many DNNs, it still has many limitations. One limitation is that it cannot explain models that deal with pairs as input such as in the SNN. As the resultant layer of SNN is the combination of two separate networks, a direct relationship between input distances does not exist, therefore, a direct implementation of LIME is not possible.

An explanation of SNN's decision route has been proposed in Utkin et al. (2021) where the feature vector is compared with the prototype of the corresponding class computed at the embedding level (the SNN output). The important features at this level are determined as features that are close to the same features of the prototype. Second, an autoencoder is trained to consider the embedding level of the Siamese network, and its decoder part is used for reconstructing input data with the corresponding changes. Numerical experiments with the well-known MNIST handwritten digits dataset illustrate the efficacy of the proposed method. The extension of this explanation method for the weakly supervised learned SNN has been discussed in Sheng and Huber (2019). The two well-known datasets, synthetic (randomly generated images by computer simulation) and MNIST, have been used to explain the explanation process of SNN. Both approaches have limitations such as the unsatisfactory reconstruction of images due to the limited dataset, and a huge number of parameters including the length of feature vectors and the number of important features. In Chen et al. (2021), the explanation for the SNN is generated by solving the constrained and unconstrained optimization problems. The gradient descent ascent and stochastic gradient descent for KL-divergence are used to solve constraint and unconstrained optimization problems respectively. The resultant invariant saliency map explains the SNN prediction. The faithfulness and counterfactual loss are evaluated on two types of datasets, tabular and graph, by the Lagrangian multiplier. This self-learn explanation technique also analyses the robustness of the explanation due

to continuous change of reference objects. Above discussed post-hoc approaches are all gradient-based and do not consider the last similarity distance layer and produce an explanation based on the encoder part of SNN.

Recently, Fedele et al. (2022) proposed a perturbation-based method, SIamese Network EXplainer (SINEX), that explains SNN outcomes for an audio dataset. SINEX uses a perturbation approach to evaluate the contribution of segment-weighted averages to the outcome. The significant contributing area of input can be visualized through heatmaps. Additionally, this paper also discussed, SINEXC, a variant of SINEX, which explains the effects of other segments on similarity scores. However, they claim that the SINEX can be implemented on single-channel grayscale images as it is but does not work on RGB images. In this paper, we attempt to address these gaps by utilising the basic concept of LIME so that we can explain the decision process from the input layer to the last distance layer of SNN. The proposed post-hoc approach provides an explanation to solve the issue of an unsatisfactory explanation of the SNN's decisions due to the limited number of samples by perturbing samples for each image. Our perturbation-based approach explains both grayscale and RGB images. In addition, the tabular information and graph based explainability highlight the correlation between the important features of image pairs.

3 Problem formulation

Lets suppose that the image dataset, $\chi =< I, Y >$ consists of N- samples $I = \{I_0, I_1, \ldots, I_{N-1}\}$ with associated labels $Y = \{Y_0, Y_1, \ldots, Y_{N-1}\}$. Each I_i is composed of either $m \times n \times 1$ (grayscale) or $m \times n \times 3$ (RGB) data arrays. Whereas each $Y_i \in [0, \ldots, E-1]$ where E is the total number of classes. In ML, the dataset is usually split into two parts: learning (training) and validation (testing).

In the learning phase, SNN model F that takes two images from $I = \{I_0, I_1, \ldots, I_N\}$ as input along with its associated labels $Y = \{Y_0, Y_1, \ldots, Y_N\}$. An image-pair passes through the twin subnetworks of SNN to transform three-dimensional data into a one-dimensional feature vector. Euclidean distance is used to find similarities between the resultant feature vectors of the image pair. The architecture of the SNN model can be shown in Fig. 1.

In this paper, we aim to solve the following problem: We have pre-trained SNN F and the objective is to establish a post-hoc explanation model G that can explain

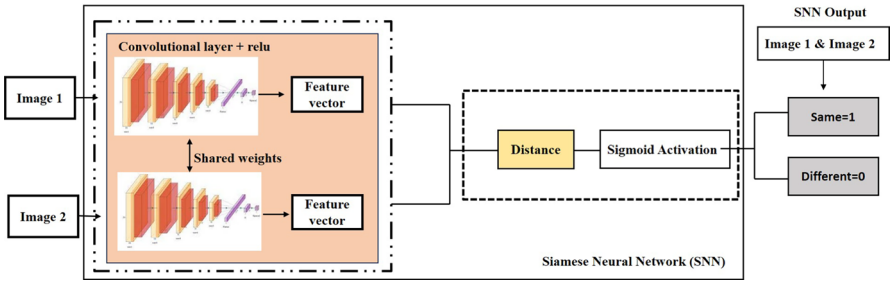


Fig. 1 The architecture of Siamese Neural Network (SNN)

the learning process within F. The proposed explanation model takes the pre-trained SNN model F and an image pair I_i, and I_j as inputs and returns an explanation ζ that discloses reasons why F has viewed these images as identical or different. Explicitly, $\zeta = G(F, I_i, I_j)$ is a collection of visual and tabular explanations for each image pair. Visualizations such as heatmaps and graphs highlight the significant features, while tabular information provides the Feature Distance-based eXplanation behind the SNN outputs.[1]

4 Proposed methodology

4.1 Features distance-based eXplanation (FDbX) method

In this section, a detailed description of the proposed post-hoc Features Distance-based eXplanation (FDbX) method for SNN has been presented along with the stability measure of our explanation approach. We divide FDbX methodology into three parts or blocks: (i) Feature block (ii) Correlation block (iii) Graph Construction and Explanation block. SNN's predictions would be explained by using the outputs of each block.

Feature block consists of steps mentioned in the first block of Fig. 2 that are used to explain the identical subnetworks of the SNN model. The identical subnetworks represent the high-dimensional images as low-dimensional feature vectors. The feature block starts the process by generating the superpixels of the given input image pairs. The goal is to represent the images into multiple segments that are more meaningful and easier to analyze. Quickshift (QS) (Garreau and Mardaoui 2021) is one of the algorithms used to segment images and used as a default algorithm to generate superpixels in LIME. QS provides a reasonable explanation for images but this algorithm does not offer a control parameter for the number of superpixels. Other prominent algorithms for superpixel generation are Felzenszwalb, Watershed, and

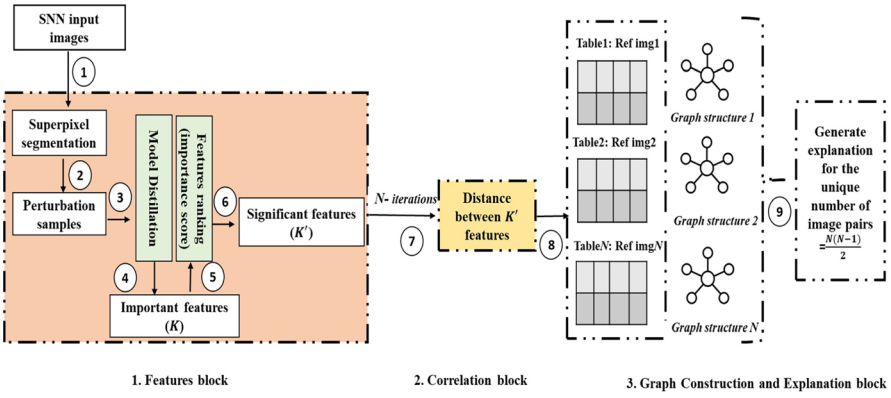


Fig. 2 Flow diagram of Features-Distance based eXplanation (FDbX) method

[1] For mathematical notations and explanation: see Appendix A.

Simple Linear Iterative Clustering (SLIC) (Schallner et al. 2020). Only SLIC offers control over the number of superpixels (Achanta et al. 2012). For this paper, the superpixels for each image have been generated by SLIC. To explain the decision route of the SNN model, we use the concept of a model distillation where a simple model is trained to imitate the complex model. In our proposed approach, we distill SNN into the ridge regression model. The prerequisite to training the ridge regression model is the availability of data samples that can be obtained by segmented images. The key to generating new data samples is to split the image into superpixels and replace these superpixels with the mean color of the superpixels. These superpixels are randomly replaced with two possibilities: on = active = 1 and off = inactive = 0. With a Bernoulli random variable of parameter $\frac{1}{2}$, each new sample consists of a set of random interpretable features that have a 50/50 chance of either being 0 or 1. This data generation technique is used in XAI techniques like LIME or SHAP since it allows the model to focus on informative regions, rather than assessing the importance of each region.

In ridge regression, a regularization term is added to the loss function of linear regression to avoid overfitting. Hence, the cost function for *N*-samples ($\{X_i, Y_i\}_{i=1}^{N}$) where each X_i has *l*-dimensional features vector can be written as.

$$\arg\min_{\beta_0,\beta}\{\frac{1}{N}\sum_{i=1}^{N}(Y_i - \beta_0 - \sum_{i=1}^{l}\beta_j X_{ij})^2\} + \alpha\sum_{j=1}^{l}\beta_j^2 \tag{1}$$

The $\beta' s$ in equation 1 are the learned feature weights, also known as coefficients. The weight β_0 without feature is called intercept. The value of $\beta' s$ can be determined by the following estimator

$$\beta = (X^T X + \alpha I)^{-1} X^T Y \tag{2}$$

The regularization coefficient α helps shrink the coefficients towards zero to improve the performance of the predictive model. Here, we chose these non-zero coefficients as *K* important features. Based on the residual sum of squares, the leave-one-out approach has been used to compute the importance of each *K* feature by removing one feature at a time from the final model. The importance score is normalized by dividing the score by the sum of all important scores, that is, $\hat{K_m} = K_m / \sum_n K_n$ to improve the integrity of the data. The sum of normalized important scores for all features should be 1 and the significance level of choosing K' significant features from *K* features is 0.05. Therefore, K' is a set of features excluding non-significant and unimportant *K* features. This step allows us to keep only those features that have the greatest impact on the SNN's decisions. Every image from the input of SNN follows these steps and explains the functionality of only the twin subnetwork part of SNN, i.e. starting from the input image pairs to the resultant feature vectors. To convert the multi-dimensional output of the convolutional layers into a one-dimensional feature vector, SNN used flatten layers. In the proposed method, after identifying K' important features, we extract those features from each image, the resultant feature vector

is of shape (number of images, K') containing only values 0 or 1. The next task is to find the correlation between these K' important features of image pairs.

Correlation block In SNN, the Euclidean distance between the two feature vectors of given input images decides whether the given image pair belongs to the same class or not. In our explanation method, we attempted to explain this decision in a correlation block. The main purpose of this correlation block is to present tabular information for each image pair. The tabular information consists of the distances between the K' features of the given image pair. Hamming, Euclidean Minkowski, and Manhattan have been commonly used distance metrics in matching problems (Suárez et al. 2021). The Hamming distance is widely used in many features matching algorithms such as the brute-force matching method ORB (Zhao et al. 2022) and works well when dealing with binary data of the same length. Hence, we select the Hamming distance to measure similarity/dissimilarity between feature vectors generated by activation(1) and inactivation(0) of superpixels. Hamming distances between vectors of equal length can range anywhere from 0 to any integer value. In this situation, we normalize the distance value by taking the ratio of the Hamming distance to the length of the vector. Mathematically, we define Hamming distance as follows:

Definition 1 Hamming distance is a metric to find the sum of corresponding elements that differ between two equal-length binary vectors say u_i and v_i.

$$\begin{aligned} &H_{dist} = \textstyle\sum_{i=1}^{k} |u_i - v_i| \geq 0 \; and \\ &H_{dist} = 0 \iff u_i = v_i \\ &\widehat{H}_{dist} = \tfrac{H_{dist}}{\|u\|} \end{aligned} \tag{3}$$

The Hamming distance between feature vectors corresponding to similar images should be smaller than the different images. We enclosed information about the feature importance of image pairs along with their Hamming distances in a tabular format to understand the relation between superpixels of input images. Next, we discuss the construction process of an explanation graph for SNN.

Graph construction and Explanation block We translate the tabular information into a raw graph to construct an 'Explanation Graph' for SNN. The important K' features from each image are the nodes of the graph and these nodes are linked (edges) by the Hamming distance *d* between them. The number of nodes and edges depends on the size and complexity of the object in the given images. It is easy to explain tiny objects such as insects with a few features, but larger or more complex images like medical images require more details to be understood (Ucer et al. 2022).

Definition 2 An undirected graph G_S is defined as a set of nodes $\aleph$ with O dimensional attributes and edges ε with S dimensional attributes in equation 4.

$$\begin{aligned} G_S &= \{\aleph, \varepsilon\} \\ \aleph &= \{\, K'_i \in \mathbb{R}^{\mathbb{O}} \;\; | \; i = 1, 2, ..., M \,\} \\ \varepsilon &= \{\, d_i \in \mathbb{R}^{\mathbb{S}} \;\; | \; K'_i, K'_j \in \aleph \,\} \end{aligned} \tag{4}$$

In Fig. 3a, the similarity between tiny insect images can be identified by a few numbers of features, say K'_i and K'_j be the two set of important features of the given image pair. Note that in Fig. 3b, we have two disjoint sets of nodes (features) K'_i and K'_j that are linked with each other by distances between them but there are no edges within the same node set. This is the type of graph known as Bipartite graph.

Now, there are ways to explain the decision-making process of SNN. Based on the feature distance-based tabular information, we can interpret why the two images are predicted as being identical by considering the diagonal values of the tabular. A bipartite undirected graph is a visual aid to understanding this explanation. The deterministic way to explain the SNN's predictions is to develop a mathematical model based on the constructed undirected bipartite explanation graph. One of our future challenges will be to develop this mathematical model. To explain the SNN outputs, there are many possible pairs to choose from the total number of images $N.E$. Where N is the number of samples for E classes. The total number of unique pairs can be calculated by the combination formula $\frac{N.E(N.E-1)}{2}$.

4.2 Stability and local fidelity of FDbX method

There is an intuitive assumption that the explanation method should adhere faithfully to the original model behaviors; otherwise, the explanation results are not acceptable. Further, the explanation method must be stable when explaining similar inputs. Due to the random sampling in the feature block of our proposed FDbX method, there is a high probability of instability in the explanation. Therefore, we discuss stability and local fidelity to evaluate the trustworthiness of our proposed FDbX method. To ensure the stability of the FDbX method, we need to define a formal description of the framework.

Consider a black box B with Q variables, a particular instance x is chosen to explain with fixed q variables. The ridge regression model g in the feature block can be defined as a mapping between the set of variables ν and corresponding coefficients $g : \nu \longrightarrow \mathbb{R}$.

1. *Stability*: Weights (coefficients) and features (variables) are vital components of our proposed explanation method. Therefore, the stability of coefficients and

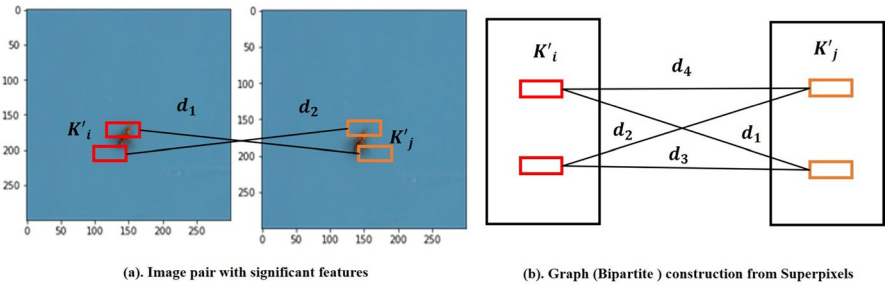


Fig. 3 Graph construction for image pairs: Superpixels are nodes and distance between nodes are edges

variables ensures the stability of the explanation. In 2020, two statistical stability indices, Variable Stability Index (VSI) and Coefficient Stability Index(CSI) (Visani et al. 2022), were introduced to check the stability of LIME but for tabular datasets. Standard Deviation (SD) of the coefficients is another stability metric for the explanation method, discussed in Shankaranarayana and Runje (2019). We present below three metrics to check the stability of the FDbX method. To our knowledge, this is the first time that VSI and CSI have been presented as stability indices for image datasets. *Standard Deviation (SD):* We iterate the *g* model for *p* times for a particular instance *x*. We use *p* coefficient values to calculate the SD of each feature and then find the average of SD for all features. Another stability metric is the SD to mean ratio. Since coefficients tend to have a wide range, they can be normalized by dividing them by the mean. To study the effect of dataset size on stability, we use these stability metrics with different numbers of generated data points to train the ridge regression model. A lower SD value would be better as it indicates that data is more clustered around the mean. *Variable Stability Index (VSI):* The variable composition of *p* generated *g* models can be computed by VSI. To calculate this index, we measure concordance among p explainable models as (g_a, g_b) pair. ν_a and ν_b represent variables of respective g_a, g_b explainable models. The mathematical formulation to check the compatibility of all possible combinations of *g* models is

$$\begin{aligned} pairs &= (g_a, g_b) \\ \nu_a &= \{feature\ \epsilon\ \nu : g_a(feature) \neq 0\} \\ \nu_b &= \{feature\ \epsilon\ \nu : g_b(feature) \neq 0\} \\ Concordance(g_a, g_b) &= |\nu_a \cap \nu_b| \end{aligned} \tag{5}$$

This equation with $Concordance(g_a, g_b)$ returns the number of those variables that are common in both g_a and g_b models. The VSI value varies from 0 to 100 as a percentage based on the average concordance value for all possible combinations. When the percentage is higher, it indicates more similar variables in *p* iterated *g* explainable models. *Coefficient Stability Index (CSI)*: To investigate the similarity between the coefficients of *p*-generated explanation models, CSI relies on the statistical distribution of coefficients and deals with confidence intervals (CI).

$$\begin{aligned} CI(g(feature)) = [&g(feature) - 1.96 \times SD(g(feature)), \\ &g(feature) + 1.96 \times SD(g(feature))] \end{aligned} \tag{6}$$

Equation (6) helps us in finding the confidence interval for all possible combinations. If the confidence intervals of different model's features are overlapped to some extent, stability is guaranteed under investigated explainable models. For this purpose, we use the binary function that takes a pair of confidence intervals $CI_{(pair)} = (CI_a, CI_b)$ as input.

$$Overlap(CI_{(pair)}) = \begin{cases} 0, & CI_a \cap CI_b = \emptyset \\ 1, & otherwise \end{cases} \tag{7}$$

Equation 7 returns value ranges from 0 to 1 based on the presence of overlap. In order to better understand the index, consider it as a percentage and scale it from 0 to 100 (Visani et al. 2022). More stability of coefficients in the neighborhood of the chosen feature can be guaranteed if CSI attains a value closer to 100.

2. *Local Fidelity*: Local fidelity describes how closely an explanation method mimics the ML model. An explanation needs to be high-fidelity, as high fidelity must have high explanation accuracy. To test how closely the FDbX method mimics the SNN model, we compute R^2 scores that explain how well the model has fit on the generated data points. We also find Mean Square Error (MSE) between local model prediction and SNN prediction to test the local fidelity of our proposed FDbX method. To find the best local fit, we compute R^2 and MSE values by varying numbers of data points and superpixels. Another parameter for comparison is to study the impact of different segmentation algorithms on the FDbX method. Therefore, we also compute R^2 and MSE for these two metrics for SLIC and QS segmentation algorithms.

5 Experimental setup

We evaluated the proposed FDbX method using a public dataset (MNIST) as well as a new dataset of western flower thrips (WFT). The MNIST is a ($28 \times 28 \times 1$) grayscale handwritten digit dataset that has 60,000 training and 10,000 testing samples.[2]The second dataset consists of images of tiny insect WFT of size 1.0 - 1.2 mm. The WFT dataset contains only 20 training and 10 testings RGB samples as SNN learns and performs well with only two to five samples i.e. few-shot learning. All images of WFT are ($300 \times 300 \times 3$) pixels.[3]

A conventional SNN consists of two identical subnetworks, for this study, the twin subnetwork is a convolutional neural network. Each subnetwork consists of three convolutional layers, max-pooling layers, and the relu activation function. The flattening layer is followed by a global average-pooling layer. Both CNNs accept three-dimensional input image pairs that are transformed into one-dimension by the flattening technique, shown in Fig. 4.

The difference between one-dimensional embedding features is calculated by L_2 Euclidean distance metric (see equation 8), which is the output of SNN (see Fig. 2).

$$L_2 = ||x_i, x_j|| = \sqrt{\sum_{i=0}^{n}(x_i - x_j)^2} \tag{8}$$

[2] Data openly available in a public athttp://yann.lecun.com/exdb/mnist/

[3] Data available on request from the authors.

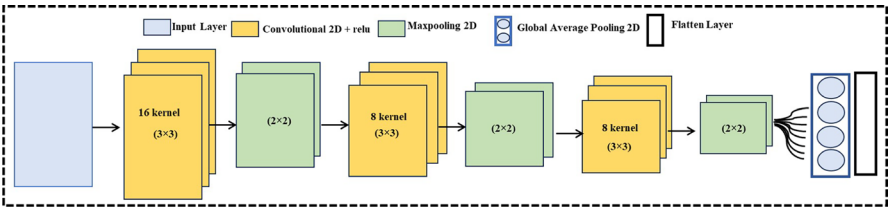


Fig. 4 The architecture of SNN's subnetworks

If two feature vectors x_i and x_j, are dissimilar, then they should be separated by a large Euclidean distance. A variety of loss functions are discussed in the article (Polela et al. 2020) with particular emphasis on minimizing or maximising distances between inputs. For example, the contrastive loss function can be defined for two feature vectors as

$$L(x_i, x_j, z_{ij}) = \begin{cases} ||h_i - h_j||_2^2 & z_{ij} = 0 \\ max(0, \phi - ||h_i - h_j||_2^2) & z_{ij} = 1 \end{cases} \tag{9}$$

Where h_i and h_j are new representations of embedding vectors. The Euclidean distance between these two embedding vectors is $z_{ij} = 0$ for similar inputs and $z_{ij} = 1$ for dissimilar inputs. Note that ϕ in the above equation is the predefined threshold that indicates that dissimilar pairs that are beyond this margin will not contribute to the loss. The *max* function is used to pick the higher value between 0 and ϕ - $||h_i - h_j||_2^2$. The total error is defined as

$$L(\Omega) = \sum L(x_i, x_j, z_{ij}) + \mu R(\Omega) \tag{10}$$

Where L is a function of Ω which is the matrix of parameters of subnetworks of SNN and μ as a hyperparameter to control the strength of regularization. Basically, equation 10 is formulated by adding the regularization term $R(\Omega)$ into the loss function (equation 9). The regularization term is added to improve the generalization of neural networks. For MNIST, we have selected learning rate= 0.01, batch-size= 16, and epochs=50 and achieved 90% training accuracy. For WFT, we have selected learning rate = 0.0006, batch-size = 5, and epochs = 100 and achieve 67% training accuracy. The learning rate, batch size, and epochs have been selected by trial and error. The explanation model's stability is affected by the accuracy of the model, so we use a random search technique to find the best hyperparameter configuration.

To implement the proposed FDbX method on the pre-trained SNN model, we used the Google Colab notebook environment for model training and testing purposes which provides free GPU access. The environment offered the Intel Xeon CPU @2.20 GHz, 13 GB RAM, Tesla K80 accelerator, and 12 GB GDDR5 VRAM and comes with preinstalled Python 3.x packages and Keras API with backend TensorFlow 2.14.0. A statistical tool IBM SPSS Statistics 28.0 has been used for assigning importance scores to features [33].

6 Results and discussion

This section discusses the results generated by the proposed FDbX method, applied to two datasets, WFT and MNIST. We present a comparison of our proposed FDbX method with the existing autoencoder (AE)-based explanation method. We also implemented the FDbX method on the CIFAR-10 dataset to explain the SNN's predicitons.[4]

6.1 FDbX for tiny insect matching

Figure 5 and 6 explain the first six steps of the proposed FDbX method. The reference image in Fig. 5 is fixed and makes pairs with the given four query images. The second and third columns show examples of segmentation and samples generated by these segmented images.

For segmentation, we used the SLIC algorithm and divided each image into 100 segments. The fourth column displays the explanation results returned by training the simple model to imitate the complex model (model distillation). The light blue areas highlight important features K for explanation while the black area indicates segments of zero influence on explanation. To ensure this, we compare our results with LIME (fifth column) which highlights the same portion of images while generating explanations. Figure 6a displays the feature weights of an image for each segmented pixel. From this, we picked the top $K = 10$ important features for the WFT. Figure 6b assigns an importance score to each K feature. The first three features (VR000041, VR000034, VR000037) have the greatest importance score of 0.969, 0.028, and 0.002 respectively. Other K features such as VR000004 have zero importance. The yellow highlighted column represents the result of the significant test that determines whether the given features are good predictors or not. Conventionally, if $p - value < 0.05$ features are statistically significant and Fig. 6b shows that only the last feature (VR000004) with zero importance is not statistically significant according to the rule. The above analysis and importance score allow us to pick $K' = 3$ features to explain the WFT images.

Based on the importance score, we compute the Hamming distance between K' features of image pairs (steps 7 and 8 in Fig. 2). Let's imagine different images $(RI_0, QI_1, QI_2, QI_3, QI_4)$ to make pairs (RI_0, QI_1), (RI_0, QI_2), (RI_0, QI_3), (RI_0, QI_4), (QI_1, QI_2), (QI_3, QI_4), (QI_1, QI_3) and (QI_2, QI_3). The distances between K' features of the four image pairs (RI_0, QI_1), (QI_1, QI_2), (QI_3, QI_4), (QI_1, QI_3) are represented in Tables 1, 2, 3, and 4 respectively.[5] The weight and importance of each feature are presented in parentheses along with the feature name. In Table 1, the first feature has 1.00 importance score, and all other features have zero importance, so only consider the distance of the feature $F30$ with the first feature $F60$ of the second image. For all other pairs, the explanation is based

[4] Supplementary file: similarity/dissimilarity between image pairs of CIFAR-10 dataset are explained by FDbX method in detail.

[5] Supplementary file: see explanation tables of image pairs (RI_0, QI_2), (RI_0, QI_3), (RI_0, QI_4) in supplementary file.

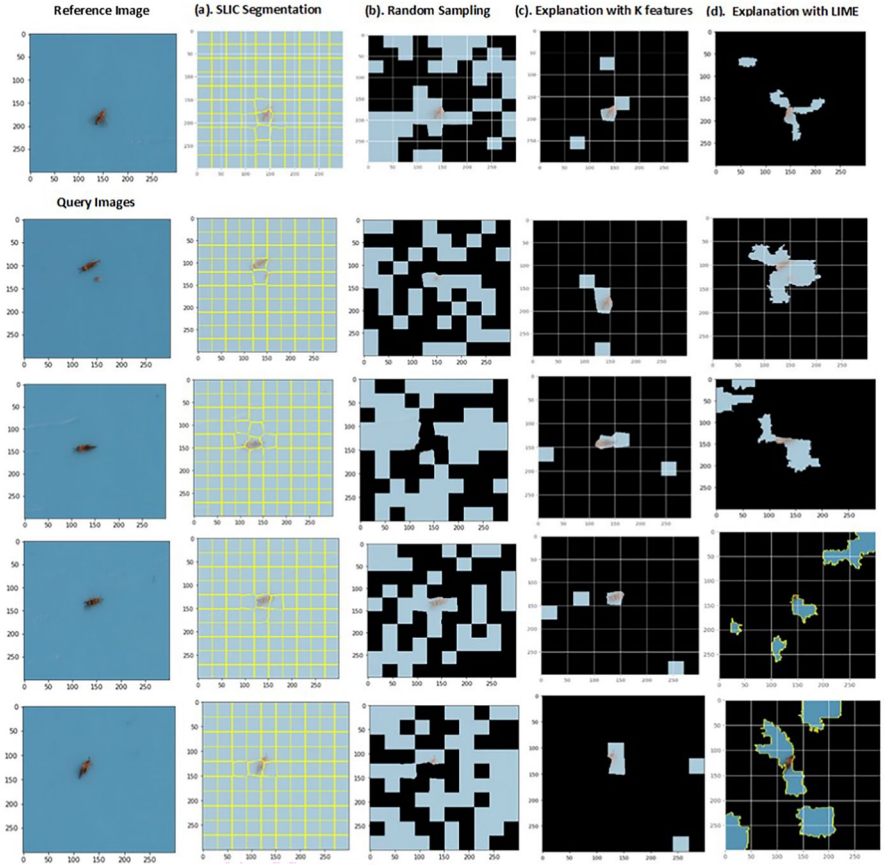
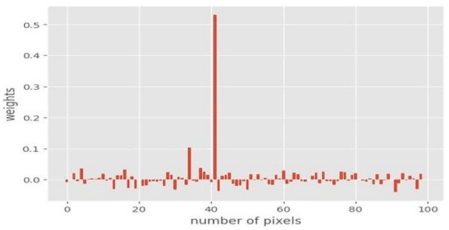


Fig. 5 Visual results of Features Block for four images: namely reference image RI_0 in the first row and query images from second to fifth row QI_1, QI_2, QI_3, QI_4. Features extraction from each image involved the following steps **a** SLIC Segmentation (second column) **b** Sampling (third column) **c** Explanation with K' features (fourth column) **d** Explanation with LIME (fifth column)



(a) Features weights vs no. of pixel

Model Term	Coefficient	Std.Error	t	Sig.	95% Confidence Interval		Importance
					Lower	Upper	
Intercept	0.114	0.009	12.067	.000	0.095	0.132	
VAR000041	0.517	0.008	61.803	.000	0.500	0.533	0.969
VAR000034	0.088	0.008	10.474	.000	0.071	0.104	0.028
VAR000037	0.026	0.008	3.089	.002	0.009	0.042	0.002
VAR00004	0.011	0.008	1.333	.185	-0.005	0.028	0.000

(b) Importance and Significance of each feature for WFT

Fig. 6 Weights and importance score of features for WFT dataset

Table 1 Weights and Importance of K' features for the image pair (RI_0, QI_1)

Feature (Weight,Importance) (RI_0)	F30 (0.78,1.00)	F52 (0.00,0.00)	F56 (0.00,0.00)	F53 (0.00,0.00)
Feature (Weight,Importance) (QI_1)				
F60 (0.49,0.70)	*2*	83	76	85
F54 (0.28,0.30)	71	*79*	74	87
F06 (0.053,0.00)	80	74	*81*	62
F96 (0.052,0.00)	62	74	67	*80*

The highlighted Hamming distances are distances between the K' important features

Table 2 Weights and Importance of K' features for the image pair (QI_1, QI_2)

Feature (Weight,Importance) (QI_1)	F60 (0.49,0.70)	F54 (0.28,0.30)	F06 (0.053,0.00)	F96 (0.052,0.00)
Feature (Weight,Importance) (QI_2)				
F40 (0.044,0.60)	*73*	71	73	71
F49 (0.041,0.35)	77	*73*	81	73
F92 (0.021,0.02)	70	68	*74*	68
F07 (0.020,0.01)	78	72	78	*66*

The highlighted Hamming distances are distances between the K' important features

Table 3 Weights and Importance of K' features for the image pair (QI_4, QI_3)

Feature (Weight,Importance) (QI_4)	F34 (0.59,0.86)	F44 (0.170,0.063)	F03 (0.022,0.007)	F51 (0.012,0.00)
Feature (Weight,Importance) (QI_3)				
F41 (0.519,0.97)	*67*	70	68	83
F34 (0.095,0.028)	72	*71*	79	74
F37 (0.035,0.002)	85	84	*70*	81
F04 (0.033,0.00)	70	77	81	*70*

The highlighted Hamming distances are distances between the K' important features

on the average distance of important features. For this study, we define threshold ϕ as $\frac{H}{2}$ if H is the feature vector's length.

Table 5 gives a compact form explanation for these image pairs. If the image pair is the same the 'True' column has a value of 1 otherwise it is 0. SNN's predictions are based on the Euclidean distance and the sigmoid function has been used to display prediction in the range [0, 1]. A value closer to 1 implies that the image pairs belong to the same class while a value closer to 0 implies they are different. The last column explains the predictions of SNNs based on our FDbX method. The threshold value for this paper is 75, the image pairs belong to different classes if the Hamming distance is greater than 75. From the above discussion, the pairs (RI_0,QI_1), (QI_1,QI_2), (RI_0,QI_3) and (RI_0, QI_4) can be declared as "same images" as they

Table 4 Weights and Importance of K' features for the image pair (QI_1, QI_3)

Feature (Weight,Importance) (QI_1)	F60 (0.49,0.70)	F54 (0.28,0.30)	F06 (0.053,0.00)	F96 (0.052,0.00)
Feature (Weight,Importance) (QI_3)				
F41 (0.519,0.97)	*86*	81	69	85
F34 (0.094,0.028)	66	*83*	69	69
F37 (0.035,0.002)	73	74	*74*	75
F04 (0.033,0.00)	79	82	86	*67*

The highlighted Hamming distances are distances between the K' important features

Table 5 True value, predictions and the explanation generated by the proposed FDbX method for different image pairs

Image pairs	True	SNN outcome = Prediction	Hamming distance = our explanation
(RI_0, QI_1)	0	0.70 = 1	72 = 1
(RI_0, QI_2)	1	0.71 = 1	74 = 1
(RI_0, QI_3)	0	0.35 = 0	87 = 0
(RI_0, QI_4)	1	0.62 = 1	78 =0
(QI_1, QI_2)	1	0.70 = 1	73 = 1
(QI_3, QI_4)	1	0.72 = 1	69 = 1
(QI_1, QI_3)	1	0.40 = 0	84 = 0
(QI_2, QI_3)	0	0.61 = 1	77 = 0

satisfy the required threshold condition and all other pairs are "different images" as the Hamming distance between image pair is more than the defined threshold. Table 5 shows that out of 8 predictions, our proposed method explains 6 predictions correctly. Next, we initialize the process of constructing an explanation graph by utilizing all the extracted information about features and the distances between them.

We discussed in Sect. 4 that graph construction is a way to explain the decision in the proposed FDbX method. Next, we will construct a graph with two independent sets of nodes without any shared edges to form a special type of graph called a Bipartite graph. Therefore, we would discuss a formal definition of a Bipartite graph before discussing how an explanation graph can be constructed for the WFT dataset.

Definition 3 A graph $G_S = \{\aleph, \varepsilon\}$ is bipartite if it has two independent sets of nodes K'_i and K'_j such as $K'_i \cup K'_j = \aleph$ and $K'_i \cap K'_j = \emptyset$ and every edge connects nodes from K'_i to K'_j or from K'_j and K'_i. With K' number of nodes bipartite graph has $\frac{1}{4} \times K'^2$ possible number of edges.

6.1.1 Construction of explanation graph

Figure 7 depicts a graph where important features of image pairs are nodes that are linked (edges) by the distance between them. The green and blue rectangles depict the first and second most important features and in the WFT dataset, these rectangles explain 90% of the target object. The white hollow rectangles represent features with zero importance. Different colors (orange, light green, purple, dark red, and light blue) of edges are used to differentiate image pairs while keeping one image fixed as the reference image.

For example, orange edges have the Hamming distance values for all image pairs (RI_0, QI_1), (RI_0, QI_2), (RI_0, QI_3) and (RI_0, QI_4) (see Table 5). Similarly, the nodes and edges for pairs (RI_1, QI_2), (RI_1, QI_3), (RI_1, QI_4) and (RI_2, QI_1), (RI_2, QI_3), (RI_2, QI_4) are mentioned in light blue and light green colour respectively. The purple edges explain the distance between pairs (RI_3, QI_1), (RI_3, QI_2), (RI_3, QI_4) and the dark red edges explains the distance for (RI_4, QI_1), (RI_4, QI_2), (RI_4, QI_3) pairs. This graph becomes wider and more complex if the number of images increases in the dataset. In Fig. 7, we try to explain the concept of the graph construction process for every image pair. Our future aim is to convert this concept into a mathematical model to generate a trustworthy deterministic explanation for SNN models. Next, to have trust in the explanations generated by the proposed method, we check the stability and fidelity of the FDbX method.

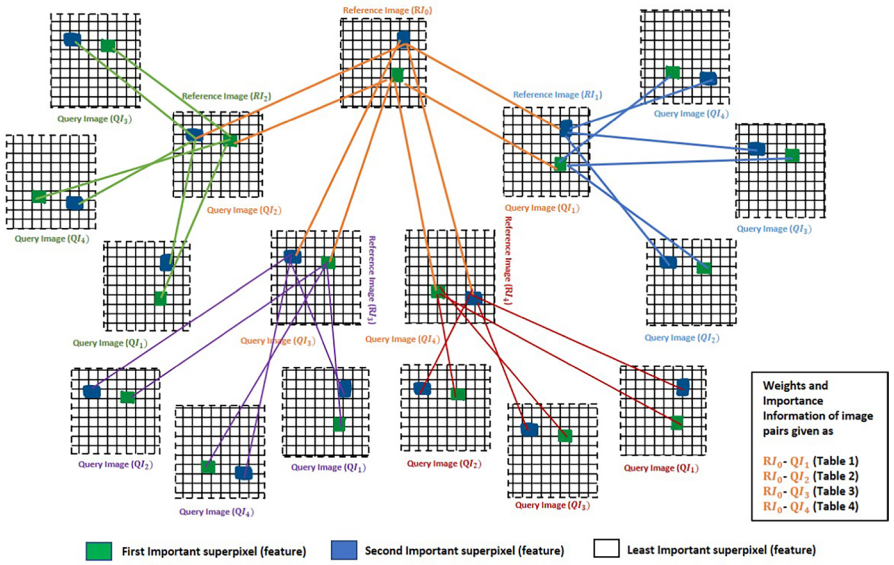
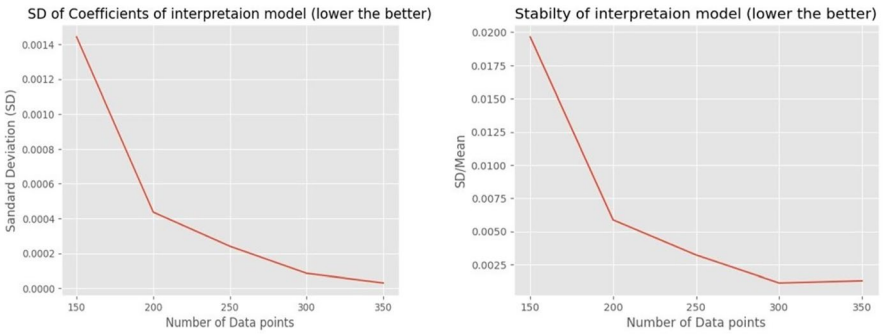


Fig. 7 The constructed explanation graph of each image pair

6.1.2 Trustworthiness of our explainability approach

In this section, We are going to discuss the trustworthiness of the FDbX method because, without trustworthiness, users may not be able to take appropriate actions based on DNN's predictions. For this, here we consider the tiny insect WFT dataset.

- *Standard Deviation (SD):* In order to measure stability, we run g explanation model $p = 10$ times and compare the standard deviations of the coefficients. First, we compute the standard deviation for each feature across ten iterations, and then we average all standard deviations. Another stability metric is to compute the SD to mean ratio (SD/mean). As coefficients tend to have varying ranges, dividing by mean serves as a form of normalization. The explanation method appears to be more stable, or consistent when the SD and SD/Mean ratio have low values. We repeat the calculation of the aforementioned two stability metrics to check the effect of different sizes of data points such as 150, 200, 250, 300, and 350. Figure 8a, 8bshow that SD and SD/mean values are decreasing with the increasing size of the dataset. This shows the stability of the FDbX method in terms of SD and SD/mean metrics.
- *CSI and VSI:* Presumably, this is the first time these indices have been used to check explanation stability for any image dataset. The process of finding VSI and CSI for an image dataset is described as follows. We take an image and all its generated samples as an instance and then calculate VSI and CSI for each generated sample. The explanation stability for that one image is the average of VSI and CSI of all generated samples. A higher percentage indicates that the same variables and coefficients have appeared for p times leading to an iterative explanation method. We compute these two indices for WFT test dataset. Figure 9 shows the satisfactory stability of the FDbX method for the WFT dataset using the CSI and VSI values.
- *Local Fidelity:* Other than stability, local fidelity is another criterion to check the reliability and trustworthiness of an explanation method. We test local fidelity



(a) Average SD (b) Ratio of SD to Mean

Fig. 8 Standard deviation as stability metrics: **a** Average SD of all features with the increasing number of data points (lower is better) **b** SD/Mean of all features with the increasing number of data points (lower is better)

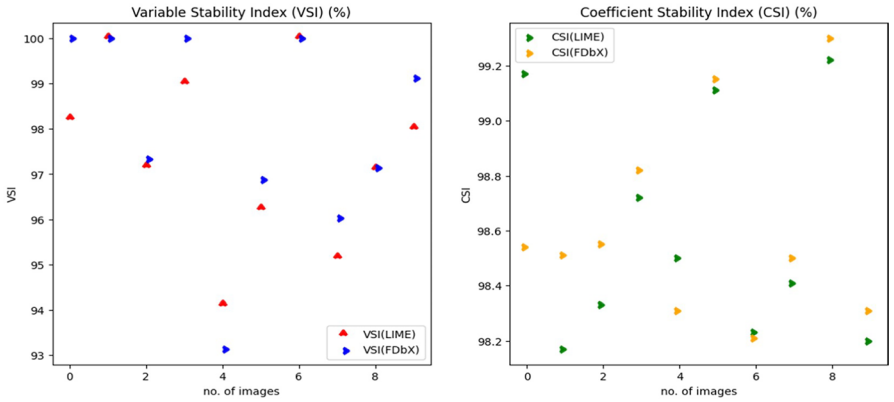
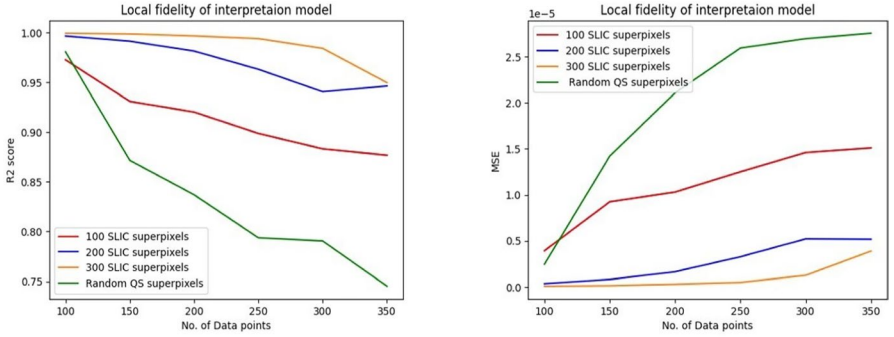


Fig. 9 VSI and CSI for different Images(range: 0–100 %) for LIME and FDbX method. High values for both indices ensure stability



(a) R^2 score vs datapoints (b) MSE vs datapoints

Fig. 10 Comparison between FDbX and LIME methods for local fidelity. R^2 score vs datapoints of a regression model with different numbers of superpixels(higher is better). MSE vs datapoints of a surrogate model with different numbers of superpixels (lower is better)

by computing R^2 scores and mean squared error (MSE) considering the different sizes of the data points. We vary the number of data points such as 100, 150, 200, 250, 300 and 350 to study the effect of the data size. It is evident from Fig. 10aand 10bthat the proposed FDbX method provides a better local fit with the SLIC algorithm. We also compare the R^2 score and MSE of our proposed FDbX method with SLIC and QS. In terms of both metrics, FDbX method with SLIC closely mimics the SNN model. In the next section, we implement the proposed FDbX method on the online available digit dataset MNIST. We illustrate the FDbX method for different pairs with different SNN outcomes. In addition, we present a comparison with one of the existing methods in the following section.

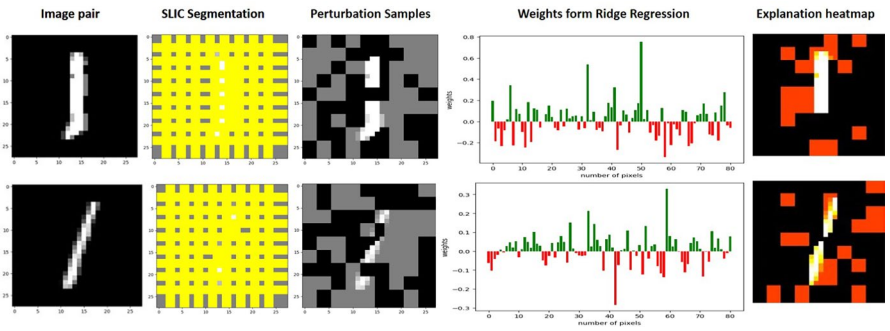


Fig. 11 Features distance-based eXplanation(FDbX) method for digit pair 1

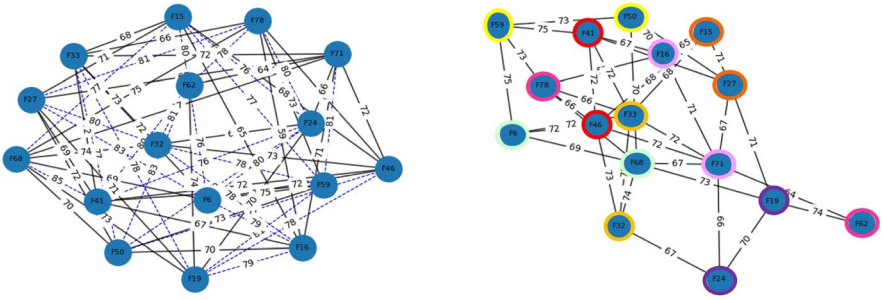
Table 6 The highlighted area are Hamming distances between the K' features for digit pair 1

Features (Importance) Digit 1 (1st row image)	F50 (0.457)	F32 (0.294)	F41 (0.060)	F19 (0.041)	F6 (0.026)	F15 (0.024)	F71 (0.021)	F78 (0.018)
Features (Importance) Digit 1 (2nd row image)								
F59 (0.330)	73	78	75	81	75	77	81	73
F33 (0.244)	77	73	72	78	72	68	72	66
F46 (0.109)	77	73	72	78	72	68	72	66
F24 (0.069)	79	67	76	70	80	76	66	80
F68 (0.055)	70	74	85	73	69	77	67	75
F27 (0.054)	72	80	69	71	83	71	67	81
F16 (0.027)	70	78	67	79	79	65	71	59
F62 (0.027)	83	81	80	74	76	80	64	78

6.2 FDbX for MNIST digit matching

MNIST is widely used as a benchmark for comparing machine learning algorithms. The dataset consists of images of ten classes of digits (from 0 to 9). Each image has an associated label that corresponds to a digit. The implementation process of the FDbX method is the same as for the WFT dataset. However, the number of important features chosen to explain SNN prediction is different in MNIST because of the size of the object i.e. digit. Initially, we select $K = 20$ important features that reduce to 8 to 10 K' features according to the assigned importance scores. Figure 11 presents results executed by the FDbX method for an image pair of the same digit 1. The heatmaps in the last column highlight important regions of both images that we chose from the third column graph. Table 6 calls attention to those features which have the highest importance scores.

Next, use the information from Table 6 to generate an undirected graph as shown in Fig. 12afor the given digit pair. The raw undirected graph is a bipartite as it is constructed with two independent sets of nodes, for digit pair 1, each set has 8 nodes. The possible number of edges in the raw undirected graph is 64. The solid black edges in Fig. 12aa is the distance less than the threshold and dotted blue edges indicate the distance greater than the threshold. Remember for MNIST, we have the same thresh-



(a) Raw Undirected Graph (b) Simplified Undirected Graph

Fig. 12 Explanation graph generated by FDbX method for digit pair 1. **a** a raw graph with all possible edges between nodes **b** a simplified graph with selected edges under the value of threshold

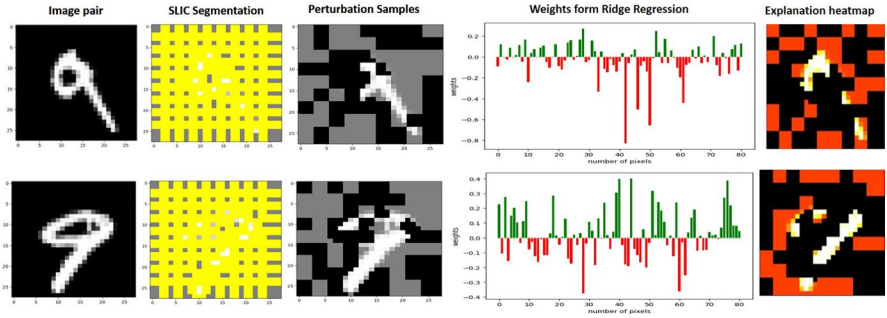


Fig. 13 Features distance-based eXplanation (FDbX) for digit pair 9

old 75. The threshold value controls the number of edges and preserves edges with distances less than the threshold. We simplified the graph by removing the edges that have a Hamming distance greater than the threshold. Hence, the graph in Fig. 12bis the simplified bipartite graph with 34 edges. We outline nodes with the same color that is on the diagonal of Table 6 to mention which node and edges we consider to explain the SNN's output. In the future, we will use this constructed graph to provide a mathematical explanation that will allow us to pick the most efficient explanation path.

Figures 13, 14 and 15 show more examples of explanations returned by the FDbX method for the MNIST dataset. Figure 13 presents a correct explanation for digit pair 9. Based on SNN, the given digit pair is predicted to belong to the same class. Our FDbX method explains the similarity between the given digit pair by explaining that the Hamming distance between important features is less than the threshold (see Table 7). On the other side, the FDbx method correctly explains that the digit 1 and digit 0 belong to different classes. Figure 14 highlights important features of different images and Table 8 explains that the distance between them is greater than the threshold. The given tabular information explains in terms of distance why SNN model predicts these images as different.

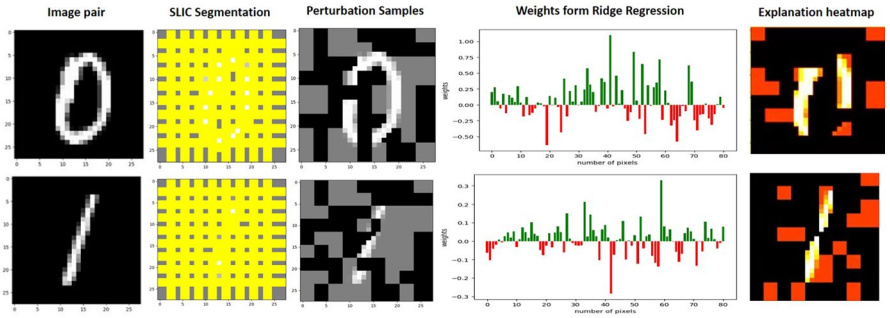


Fig. 14 Features distance-based eXplanation (FDbX) for digit pair 0 and 1

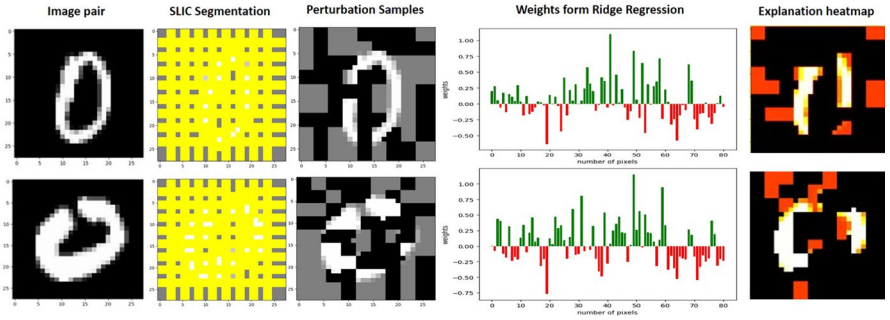


Fig. 15 Features Distance-based eXplanation (FDbX) for digit pair 0

Another interesting case is when the same digit from MNIST is incorrectly explained by the proposed FDbX method. This case is demonstrated in Fig. 15 and Table 9. The SNN predicts that the given images of digit 0 belong to the same class but the distance information given in tabular format indicates that the given image pair is different. Hence, the proposed FDbX method incorrectly explains the SNN's prediction for digit 0.

In this section, we illustrate four examples from MNIST of which three are correctly explained by the proposed FDbX method. Next, we present a comparison of our method with the existing autoencoder(AE)-based method for the MNIST dataset.

In the next section, we compare our proposed FDbX method with the existing XAI methods: LIME and Autoencoder-based SNN explanation method. The first Sect. 7.1 compares our explanation with the LIME method on the WFT dataset. In the second Sect. 7.2, we present an explanation comparison of the SNN's predictions on the MNIST dataset with the proposed FDbX method and Autoencoder-based explanation.

Table 7 The highlighted area are Hamming distances between the K' features for digit pair 9

Features (Importance) Digit 9 (1st row image)	F31 (0.142)	F4 (0.120)	F9 (0.101)	F28 (0.096)	F24 (0.085)	F45 (0.078)	F78 (0.071)	F7 (0.052)	F34 (0.040)	F14 (0.030)
Features (Importance) Digit 9 (2nd row image)										
F44 (0.132)	67	74	81	77	72	77	67	64	76	84
F76 (0.118)	77	69	79	79	74	71	73	74	68	66
F40 (0.105)	87	82	71	79	80	79	71	72	78	58
F51 (0.067)	77	78	73	69	70	81	67	82	72	78
F2 (0.050)	75	68	77	71	74	71	75	72	74	72
F18 (0.049)	71	70	81	77	77	69	85	66	82	64
F39 (0.047)	75	76	79	73	73	73	67	64	72	80
F9 (0.046)	66	67	72	88	81	82	84	81	79	77
F75 (0.044)	80	77	70	72	73	64	72	71	63	75
F54 (0.042)	74	83	70	82	63	78	66	69	69	65

7 Comparison with the existing XAI method

7.1 Comparison with the existing LIME method

As can be seen in Fig. 5, the fourth column represents the explanation generated by the proposed FDbX method using 100 superpixels and $K' = 4$ important features for the WFT dataset. Whereas the last column shows the LIME explanation for each image. Using Python's LIME package for images, the quickshift segmentation algorithm generates the random number of superpixels (greater than 100) for the given images. Each image takes $K' = 8, 9, 8, 10$, and 15 important features respectively to give an adequate explanation by LIME. We further compare the trustworthiness of LIME with the FDbX method in Sect. 6.1.2. We run both explanation methods $p = 10$ times for each image to check the coefficients and variable stability of LIME and FDbX methods. Based on the stability indices (see Fig. 9), our proposed method is more stable for most images than LIME. In Fig. 10, we can observe that the FDbX method (using SLIC) is locally more faithful than LIME (using QS).

7.2 Comparison with the existing autoencoder-based explanation method

We compare the proposed FDbX method with the existing autoencoder(AE)-based explanation method of SNN for the publically available MNIST dataset (Utkin et al. 2021). In the AE-based method, the feature vector is compared with the prototype of the corresponding class computed at the embedding level (the SNN output). The important features at this level are determined as features that are close to the same features of the prototype. Second, an autoencoder is trained to consider the embedding level of the Siamese network, and its decoder part is used for reconstructing input data with the corresponding changes. Essentially, AE-based explanation method is trained as a separate autoencoder (AE) for each image of SNN and the decoder part reconstructs the image corresponding to the subnetwork.

If the size of the dataset is small the reconstruction of the image using the decoder part is not satisfactory, hence the authenticity of the AE-based explanation method is compromised. In the comparison, the availability of limited data may not affect our explanation as we perturb samples before generating an explanation for each instance. Heatmaps in Fig. 16 show the explanation of the MNIST dataset with the different number of important features. The first column is actual image pairs that are predicted as the same class images. The second and third columns highlighted the important 10 and 12 features that FDbX method highlighted while generating the explanation. Using the same number of features, the AE-based explanation reconstructs the images. The heatmaps of FDbX method with the $K\prime$ features give a more clear explanation for MNIST than the existing autoencoder (AE)-based method. In addition, we provide a tabular and graphical explanation based on the $K\prime$ feature distance. The notation and mathematical expressions used in this paper are listed in Table 10.

Table 8 The highlighted area are Hamming distances between the K' features for digit pair 0 and 1

Features (Importance) Digit 0 (1st row image)	F41 (0.240)	F49 (0.173)	F68 (0.124)	F52 (0.112)	F58 (0.087)	F33 (0.050)	F43 (0.040)	F56 (0.036)	F38 (0.026)
Features (Importance) Digit 1(2nd row image)									
F59 (0.330)	76	80	71	73	73	77	75	77	84
F33 (0.244)	72	77	78	82	68	70	82	72	79
F46 (0.109)	72	75	78	82	68	70	82	72	79
F24 (0.069)	76	71	60	84	80	64	74	80	69
F68 (0.055)	85	82	81	73	81	73	70	79	70
F27 (0.054)	69	76	87	75	83	67	79	71	70
F15 (0.027)	67	82	77	79	77	83	79	71	78
F62 (0.027)	80	83	76	62	74	84	60	84	65
F74 (0.026)	80	73	78	80	68	64	80	74	79

8 Conclusions and future directions

In this paper, we proposed a novel method called Features Distance-based eXplanation (FDbX) to interpret the decisions made by Siamese Neural Networks (SNNs) by correlating important features from a given input image pair. To generate explanations, we employed a model distillation technique that trains a simpler surrogate model to extract and highlight the features contributing to the predictions of the more complex SNN. By assigning importance scores to the extracted features, our method focuses attention on the most relevant aspects of the input that influence the decision-making process. Since SNNs base their decisions on the distance between feature vectors, our approach mimics this behavior by computing the Hamming distance—but only between the most important features. These features and their distances are represented as nodes and edges in an undirected bipartite explanation graph. A threshold is applied to filter out less relevant edges, transforming the dense graph into a sparse one that captures only the most critical features. This not only enhances interpretability but also reduces the computational cost of model training. We demonstrate that a few high-precision matches are sufficient to explain a learning decision effectively. The implementation of FDbX on two datasets shows that our method provides satisfactory explanations of SNN predictions. Through extensive experimentation, we confirmed that FDbX achieves both local fidelity and stability, and significantly outperforms existing autoencoder-based approaches in terms of visual clarity and interpretability.

In future research, we aim to enhance the representation of the FDbX method to identify the most efficient explanation paths and improve confidence in the interpretability of SNNs. One direction involves transforming the bipartite graph into a signed bipartite network, where edges are labeled as positive or negative based on feature relevance. This will allow the method to focus only on highly important nodes with positive edge labels, further reducing computational complexity. We also plan to explore mathematical approaches to feature extraction, such as using wavelet transforms instead of training a surrogate model for each instance. Wavelet transforms decompose an image into coefficients that represent energy distribution

Table 9 The highlighted area are Hamming distances between the K' features for digit pair 0

Features (Importance) Digit 0(1st row image)	F41 (0.240)	F49 (0.173)	F68 (0.124)	F52 (0.112)	F58 (0.087)	F33 (0.050)	F43 (0.040)	F56 (0.036)	F38 (0.026)	F69 (0.026)
Features (Importance) Digit 0(2nd row image)										
F49 (0.233)	76	77	76	74	82	82	72	74	73	83
F39 (0.136)	72	87	70	78	72	72	72	82	79	67
F59 (0.116)	81	88	67	79	73	69	69	75	82	74
F31 (0.068)	76	79	82	68	76	70	80	70	79	75
F53 (0.062)	74	63	82	80	80	80	80	74	69	85
F22 (0.055)	77	70	73	73	69	81	65	83	76	82
F44 (0.040)	85	78	65	77	75	71	83	73	70	76
F24 (0.039)	69	76	79	71	75	69	69	73	64	72
F11 (0.033)	81	72	85	73	79	71	81	65	64	82
F28 (0.033)	78	75	74	72	76	70	76	70	75	73

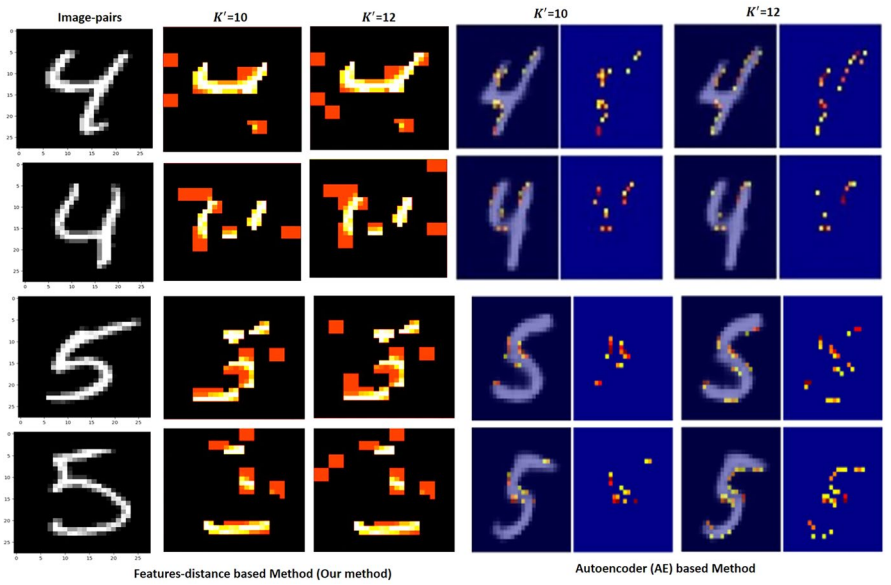


Fig. 16 Visual comparison of FDbX method (second and third columns) and autoencoder (AE)-based (from fourth to seven columns) explanation of Digit 4 and 5 from MNIST dataset for K'=10 and 12

across different scales and frequencies. These coefficients offer a compact and robust representation of the image, invariant to translation, rotation, and scaling, and can be used to isolate significant features. Additionally, we will expand our experimental setup to apply FDbX to more complex neural networks trained on sensitive medical datasets, such as cardiovascular scans collected from real-world hospital environments. The use of explainable AI in medical applications is crucial for ethical, safe, and transparent decision-making. It strengthens doctor-patient trust and helps identify biases in models—especially those arising from sensitive attributes like race, gender, or socioeconomic status. XAI can play a vital role in mitigating these biases by revealing how such features influence predictions, ensuring fairness in healthcare outcomes. While methods like Grad-CAM and attention mechanisms have successfully explained image classification, object detection, and semantic segmentation, XAI remains underexplored in the context of image similarity analysis. Developing explainable SNNs is essential not only for improving the usability of similarity-based applications but also for helping users understand the reasoning behind the results. In medical imaging, similarity-based comparisons can support the diagnosis and identification of diseases and anomalies, making explainable SNNs a valuable tool in clinical decision support systems.

Table 10 Section-wise mathematical notations used in the paper

Sections	Notation with explanation
Section 3 (problem formulation)	χ= Image dataset I= Collection of images Y= Collection of labels F= Pre-trained SNN model G= Explanation method ζ= Collection of explanations E= Classes in dataset
Section 4 (proposed methodology)	N= Number of samples l= dimension of features vector u_i, u_j= Binary vectors H_{dist}= Hamming distance K= Important features K'= SPSS selected important features x and y = feature and target of regression model β= Feature weight in regression model α= Regularization coefficient B= Black box Q= Collection of variables p= Iterations g= Regression model q= Fixed variable ν= Set of variables G_S= Undirected graph constructed by segmentation $\aleph$=Set of graph nodes ε= Set of graph edges d_i=Distance between graph nodes
Section 5 (experimental setup)	L_2= Euclidean distance x_i, x_j= Feature vectors h_i, h_j= Embedding vectors z_{ij}= Euclidean distance between two embedding vectors ϕ= predefined threshold for Euclidean distance Ω= Error function R= Regularization term μ= Control hyperparameter
Section 6 (results and discussion)	$\|\|u\|\|$= Length of feature vector RI= Reference image QI= Query Image

Mathematical notations

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s10618-025-01154-y.

Author contributions Rabia Saleem: Conceptualization, Methodology and investigation, Writing - original draft preparation and editing Ashiq Anjum: Supervision, Critical review methodology, Writing - review and editing Bo Yuan: Supervision, Writing - review Lu Liu: Writing - review

Data availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declare no Conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

Leventi-Peetz A-M, Östreich T (2022) Deep learning reproducibility and explainable ai (xai). arXiv preprint arXiv:2202.11452

Roy SK, Harandi M, Nock R, Hartley R (2019) Siamese networks The tale of two manifolds In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3046–3055

Bromley J, Guyon I, LeCun Y, Säckinger E, Shah R (1993) Signature verification using a" siamese" time delay neural network. Adv Neural Inf Process Syst 6

Bromley J, Bentz JW, Bottou L, Guyon I, Lecun Y, Moore C, Sackinger E, Shah R (1993) Signature verification using a siamese time delay neural network. Int J Pattern Recognit Artif Intell 7(4)

Zhao L, Yang J, Zhang Y, Huang J (2022) Research on feature matching of an improved orb algorithm. In 2022 IEEE 6th Information Technology and Mechatronics Engineering Conference (ITOEC) 6:765–769. https://doi.org/10.1109/ITOEC53115.2022.9734583

Wilming R, Budding C, Müller K-R, Haufe S (2022) Scrutinizing xai using linear ground-truth data with suppressor variables. Mach Learn 111(5):1903–1923

Kadir MA, Mosavi A, Sonntag D Assessing xai: Unveiling evaluation metrics for local explanation, taxonomies, key concepts, and practical applications

Garreau D, Mardaoui D (2021) What does lime really see in images? In: International Conference on Machine Learning, pp. 3620–3629. PMLR

Utkin L, Kovalev M, Kasimov E (2021) An explanation method for siamese neural networks. In: Proceedings of International Scientific Conference on Telecommunications, Computing and Control: TELECCON 2019, pp. 219–230. Springer

Sheng T, Huber, M (2019) Siamese networks for weakly supervised human activity recognition. In: 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), pp. 4069–4075. https://doi.org/10.1109/SMC.2019.8914045

Fedele A, Guidotti R, Pedreschi D (2022) Explaining siamese networks in few-shot learning for audio data In: Discovery Science: 25th International Conference, DS 2022, Montpellier, France, October 10–12, 2022, Proceedings, pp. 509–524 . Springer

Schallner L, Rabold J, Scholz O, Schmid U (2020) Effect of superpixel aggregation on explanations in lime-a case study with biological data In: Machine Learning and Knowledge Discovery in Databases: International Workshops of ECML PKDD 2019, Würzburg, Germany, September 16–20, 2019, Proceedings, Part I, pp. 147–158. Springer

Rong H-g, Li Y-j, Zhang X-s (2022) An image matching algorithm based on bipartite graph. In: ISCTT 2022; 7th International Conference on Information Science, Computer Technology and Transportation, pp. 1–4. VDE

Caro-Martínez M, Jiménez-Díaz G, Recio-Garcia JA (2023) A graph-based approach for minimising the knowledge requirement of explainable recommender systems. Knowl Inf Syst 1–31

Visani G, Bagli E, Chesani F, Poluzzi A, Capuzzo D (2022) Statistical stability indices for lime: obtaining reliable explanations for machine learning models. J Operation Res Soc 73(1):91–101

Saleem R, Yuan B, Kurugollu F, Anjum A, Liu L (2022) Explaining deep neural networks: A survey on the global interpretation methods. Neurocomputing

Antwarg L, Miller RM, Shapira B, Rokach L (2021) Explaining anomalies detected by autoencoders using shapley additive explanations. Expert Syst Appl 186:115736
Chakraborty T, Trehan U, Mallat K, Dugelay J-L (2022) Generalizing adversarial explanations with grad-cam. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 187–193
Riaz H, Smeaton AF (2023) Vision based machine learning algorithms for out-of-distribution generalisation arXiv preprint arXiv:2301.06975
Ribeiro MT, Singh S, Guestrin C (2016) " why should i trust you?" explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1135–1144
Zafar MR, Khan NM (2019) Dlime A deterministic local interpretable model-agnostic explanations approach for computer-aided diagnosis systems arXiv preprint arXiv:1906.10263
Rabold J, Deininger H, Siebers M, Schmid U (2020) Enriching visual with verbal explanations for relational concepts-combining lime with aleph In: Machine Learning and Knowledge Discovery in Databases: International Workshops of ECML PKDD 2019, Würzburg, Germany, September 16–20, 2019, Proceedings, Part I, pp. 180–192. Springer
Sangroya A, Rastogi M, Anantaram C, Vig L (2020) Guided-lime Structured sampling based hybrid approach towards explaining blackbox machine learning models In: CIKM (Workshops)
Shi S, Zhang X, Fan W (2020) A modified perturbed sampling method for local interpretable model-agnostic explanation arXiv preprint arXiv:2002.07434
Bramhall S, Horn H, Tieu M, Lohia N (2020) Qlime-a quadratic local interpretable model-agnostic explanation approach. SMU Data Sci Rev 3(1):4
Gaudel R, Galárraga L, Delaunay J, Rozé L, Bhargava V (2022) s-lime Reconciling locality and fidelity in linear explanations In: Advances in Intelligent Data Analysis XX: 20th International Symposium on Intelligent Data Analysis, IDA 2022, Rennes, France, April 20–22, 2022, Proceedings, pp. 102–114. Springer
Chen C, Shen Y, Ma G, Kong X, Rangarajan S, Zhang X, Xie S (2021) Self-learn to explain siamese networks robustly In: 2021 IEEE International Conference on Data Mining (ICDM), pp. 1018–1023. IEEE
Achanta R, Shaji A, Smith K, Lucchi A, Fua P, Süsstrunk S (2012) Slic superpixels compared to state-of-the-art superpixel methods. IEEE Trans Pattern Anal Mach Intell 34(11):2274–2282
Suárez JL, García S, Herrera F (2021) A tutorial on distance metric learning: Mathematical foundations, algorithms, experimental analysis, prospects and challenges. Neurocomputing 425:300–322
Ucer S, Ozyer T, Alhajj R (2022) Explainable artificial intelligence through graph theory by generalized social network analysis-based classifier. Sci Rep 12(1):15210
Shankaranarayana SM, Runje D (2019) ALIME: autoencoder based approach for local interpretability. CoRR **abs/1909.02437**
Polela JA et al (2020) The influence of loss function usage at siamese network in measuring text similarity. Int J Adv Comput Sci Appl 11(12)
IBM Corp.: IBM SPSS Statistics for Windows. https://hadoop.apache.org

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Rabia Saleem[1] · Ashiq Anjum[1] · Bo Yuan[1] · Lu Liu[1]

✉ Rabia Saleem
rs802@leicester.ac.uk

✉ Ashiq Anjum
a.anjum@leicester.ac.uk

Bo Yuan
b.yuan@leicester.ac.uk

Lu Liu
l.liu@leicester.ac.uk

[1] School of Computing and Mathematical Sciences, University of Leicester, University Rd, Leicester LE1 7RH, UK