INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

ORIGINAL ARTICLE

Enhancing Generalized Autoencoder for Improved Supervised Dimensionality Reduction in Image Recognition

Purvi A Koringa[1]*, Nisha K Prajapati[1], Hina K Patel[1], Pooja C Gajjar[1], Nehal M Shyal[1]

1 Electronics and Communication Engineering Department, Government Engineering College, Gandhinagar, Gujarat, India

OPEN ACCESS

Received: 14/08/2025

Accepted: 26/09/2025

Published: 20/10/2025

Citation: Koringa PA, Prajapati NK, Patel HK, Gajjar PC, Shyal NM (2025) Enhancing Generalized Autoencoder for Improved Supervised Dimensionality Reduction in Image Recognition. Indian Journal of Science and Technology 18(37): 3061-3072. https://doi.org/10.17485/IJST/v18i37.1422

* **Corresponding author**.

purvi.koringa@gmail.com

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Objectives: The article proposes enhancements to the Generalized Autoencoder (GAE) framework for Dimensionality Reduction (DR).The first objective is to improve the reconstruction process so that recognition accuracy increases. The second objective is to embed class label information into the framework. This turns GAE into a supervised method with better interclass discrimination. Method: Two improvements to the GAE are proposed: (i) an adjustment to reconstruction weights to capture neighbourhood relations in high-dimensional spaces effectively, and (ii) a strategy that incorporates class label information directly into the reconstruction set. The approach is evaluated using standard benchmark datasets for face recognition and handwritten digit classification. The approach is evaluated on eight benchmark datasets: four standard face databases (ORL, AR, UMIST, YaleB) and four handwritten digit databases (MNIST, Gujarati, Devnagari, Bangla). For a fair comparison, recognition performance is studied under varying dimensionality of the reduced feature space, different neighborhood sizes. Findings: Experimental results show that the modified GAE significantly improves recognition performance compared to the original GAE by 1-5%. The supervised version demonstrates stronger discriminative power across all datasets with better recognition rates by 3-7%. Novelty: The proposed approach adds class label information into the reconstruction set, turning GAE into a supervised framework. At the same time, it refines the weight adjustment process. Together, these two changes improve both accuracy and discrimination in dimensionality reduction tasks.

Keywords: Dimensionality Reduction; Auto Encoder; Laplacian EigenMaps; Locally Linear Embeddings; Locality Preserving Projection; Neighbourhood Preserving Projection

1 Introduction

In practical scenarios, data used in machine learning, data mining, and computer vision tasks often exist in very high-dimensional spaces, for instance- facial images, handwritten digits, and textual documents. Working with such high-dimensional

datasets introduces challenges commonly referred to as the "*curse of dimensionality*," particularly in classification and pattern recognition tasks. However, such data usually has limited degrees of freedom, and their distribution often follows a low-dimensional manifold. Identifying this manifold has therefore become a fundamental problem in the machine learning field. Many Dimensionality Reduction (DR) methods have been proposed and successfully applied in areas such as pattern recognition, data visualization, and information retrieval.

Principal Component Analysis (PCA) is a classical and widely used DR technique. It projects data in the direction of maximum variance and thus preserves global variance. However, PCA is unsupervised and does not utilize class information. Supervised linear methods such as Linear Discriminant Analysis (LDA) and Marginal Fisher Analysis (MFA) reduce intra-class variance and enhance inter-class separability in the projected space, but like PCA, they are linear methods and fail when data lies on nonlinear manifolds. To address this, methods like Isometric Mapping (ISOMAP) preserve geodesic distances of high-dimensional data points while embedding them in lower-dimensional space. Laplacian Eigenmaps (LE) further preserve pairwise similarities but lack an explicit mapping for unseen test samples, a limitation that was addressed by Locality Preserving Projection (LPP). Locally Linear Embedding (LLE) learns manifold structures by preserving local neighborhoods but also suffers from the out-of-sample problem, which Neighbourhood Preserving Projections (NPP) attempt to resolve.

With the rise of neural representation learning, autoencoders[1] (AEs) have become popular dimensionality reduction (DR) methods. A linear AE is equivalent to PCA, but in general AEs are trained by minimizing the reconstruction error between inputs and outputs and thus fail to model inter-data relationships. To address this, recent works have introduced manifold-aware extensions of AEs. For example, the Deep Laplacian Autoencoder integrates Laplacian regularization into the AE objective to smooth manifold structures and was applied successfully in fault diagnosis of rotating machinery[2]. Similarly, a topology-preserving AE that incorporates local distance and neighborhood topology preservation into the learning objective, thereby improving representation of manifold geometry, has been proposed[3]. Other works have evaluated AE-based DR in practical pipelines; for instance, a systematic analysis of AE-driven DR for face recognition tasks highlighted both strengths and limitations in realistic scenarios[4]. Autoencoders have also been applied for denoising and dimensionality reduction in image processing[5], IoT image data analysis[6], and have been extensively surveyed in recent reviews that provide new perspectives on deep autoencoder architectures[7]. While these studies confirm the versatility of AE-based DR, they largely remain unsupervised and focus primarily on minimizing reconstruction error, without directly leveraging class information to enhance discriminative power.

Generalized Autoencoders (GAE) have been proposed as a solution by embedding data relations into the AE framework through graph embedding. Unlike conventional AEs, GAEs explicitly incorporate neighborhood relations, enabling them to capture both local and global manifold structures. Studies have shown their effectiveness in manifold learning and their ability to unify AE with classical graph-based DR methods. However, existing GAEs remain unsupervised and do not exploit class label information, which is crucial for improving discriminative ability in supervised learning scenarios. This motivates the present work, which aims to enhance GAE variants—specifically GAE-LE and GAE-LLE—by (i) introducing a modified weighting scheme that improves local manifold preservation and recognition accuracy, and (ii) incorporating class label information directly into the GAE objective to boost supervised performance. Experiments on four face datasets and four handwritten digit datasets demonstrate that the proposed methods consistently outperform state-of-the-art DR techniques, establishing their efficacy in both unsupervised and supervised settings.

2 Methodology

For given data, let $\mathbf{X} \in \mathbb{R}^{m \times M}$ be the data matrix in high dimension where each column is an input data vector denoted as $\mathbf{x}_1$, $\mathbf{x}_2$, ..., $\mathbf{x}_M$; $\mathbf{Y} \in \mathbb{R}^{d \times M}$ be the low dimensional representation of $\mathbf{X}$, where each column is output data vector denoted as $\mathbf{y}_1$, $\mathbf{y}_2$, ..., $\mathbf{y}_M$. Here $d << m$. Non-linear dimensionality reduction methods usually find implicit mapping $\mathcal{F}: \mathbf{X} \rightarrow \mathbf{Y}$. Whereas, linear DR methods give explicit mapping from $\mathbf{X}$ to $\mathbf{Y}$ using a projection matrix $\mathbf{V}$ usually related as $\mathbf{Y} = \mathbf{V}^T \mathbf{X}$.

2.1 Nonlinear Dimensionality Reduction

2.1.1. Laplacian Embeddings

Laplacian Embeddings[8] (LE) uses a graph embedding approach to find underlying low dimensional manifold of the data. Each data is considered a vertex of an undirected neighbourhood. If datapoint $\mathbf{x}_j$ is among the k-nearest neighbours of $\mathbf{x}_i$, they are connected by an edge with penalty $\mathbf{P}_{ij} = 1$, and vice-versa. i.e. $\mathbf{x_j} \in N_i$ or $\mathbf{x_i} \in N_j$; otherwise, they are not connected by edge. This binary weighting method works well in practice. However, different weighting schemes are used for better performance based on the application at hand.

This approach seeks a low-dimensional representation by placing neighbouring data points as close together as possible. At the same time, it does not give any emphasis to the data points that are not neighbours. The optimization cost function of LE is given as:

$$\text{argmin}_{i,j} \left\|\mathbf{y}_i - \mathbf{y}_j\right\|^2 \mathbf{P}_{ij} \tag{1}$$

Here, **P** is a penalty matrix that represents the neighbourhood relationship in higher dimension. **Y** are low dimensional embeddings of data **X**.

The closed-form solution to the objective function (**Eq. 1**) results in a generalized eigenvalue problem of the form **LY** = λ**DY**, where **L = D - P** represents the graph Laplacian, and **D** is a diagonal matrix with entries $\mathbf{D}_{ii} = \sum_j \mathbf{P}_{ij}$. The low-dimensional embeddings are obtained from the eigenvectors corresponding to the smallest *d* eigenvalues of this problem.

2.1.2. Locally Linear Embeddings

Locally Linear Embedding[9] (LLE) learns low-dimensional manifold structure by considering local neighbourhood patches as linear patches and finds global low-dimensional representation that best preserve this linearity. Suppose each data point $\mathbf{x}_i$ has a small set of neighbour data points in **X**. Let this neighbourhood set be N_i. This small neighbourhood patch can be considered locally linear, and thus the data point $\mathbf{x}_i$ can be reconstructed as a linear combination of its neighbours, expressed as $\sum_{j_{\in N_i}} w_{ij}\mathbf{x}_j$

The weights w_{ij} can be found by solving the following optimization problem. The weights $w_{ij} = 0$ if $\mathbf{x}_j$ is not a neighbour of $\mathbf{x}_i$.

$$\underset{i}{argmin} \|\mathbf{x_i} - \textstyle\sum_{j\in N_i} w_{ij}\mathbf{x_j}\|^2 \tag{2}$$

In the next step, the low-dimensional embeddings y_i are computed such that the neighbourhood relationships of each data point with its neighbours are preserved using the same weights as in the high-dimensional space. The cost function for the LLE (Locally Linear Embedding) projections is given by:

$$argmin \textstyle\sum_{i=1}^{M} \left\|\mathbf{y_i} - \sum_{j=1}^{M} w_{ij}\mathbf{y_j}\right\|^2 \tag{3}$$

2.2 Dimensionality Reduction using Neural Network

2.2.1. Autoencoder:

Autoencoder learn efficient low dimensional data representation in an unsupervised manner. It is a three-layer feed-forward neural network designed to replicate its input at the output layer by minimizing the reconstruction error during training, while the hidden layer usually compresses the information in what is known as encoding or latent variable. Figure 1(a) shows the architecture of an autoencoder. These deep autoencoders use Restricted Boltzmann Machine to pre-train the network that can approximate a good solution, then use backpropagation to fine-tune the results.

In an autoencoder (Figure 1(a)), the encoder part consists of an input layer and several hidden layers; the decoder part consists of several hidden layers and output layer. The encoder maps input $\mathbf{x_i} \in \mathbb{R}^{\mathrm{m}}$ to the compressed representation $\mathbf{y_i} \in \mathbb{R}^{\mathrm{d}}$ by function $f(\cdot)$:

$$\mathbf{y_i} = f(\mathbf{W}\mathbf{x_i}) \tag{4}$$

$f(\cdot)$ is an activation function. If it is a sigmoid function, the autoencoder yields a nonlinear projection. If $f(\cdot)$ is identity, then the autoencoder yields a linear projection. Here, **W** is a $d \times m$ hyper-parameter matrix. For simplicity of notation, bias terms are ignored.

$$\mathbf{x_i}' = g(\mathbf{W}^T\mathbf{y_i}) \tag{5}$$

Here, $\mathbf{W}^{\mathrm{T}}$ is an $m \times d$ hyper-parameter matrix. $g(\cdot)$ is an activation function. **W** and $\mathbf{W}^{\mathrm{T}}$ are trained by minimizing reconstruction loss given as:

$$E(\mathbf{x_i}, \mathbf{x_i}') = \textstyle\sum_{i=1}^{M} \|\mathbf{x_i} - \mathbf{x_i}'\|^2 \tag{6}$$

Deep networks are expected to give better compression as compared to shallow autoencoders. These kinds of neural networks and their extensions are successfully employed to learn meaningful features of data[10]. However, these methods only focus on individual data points. Thus, it fails to model relations among data points.

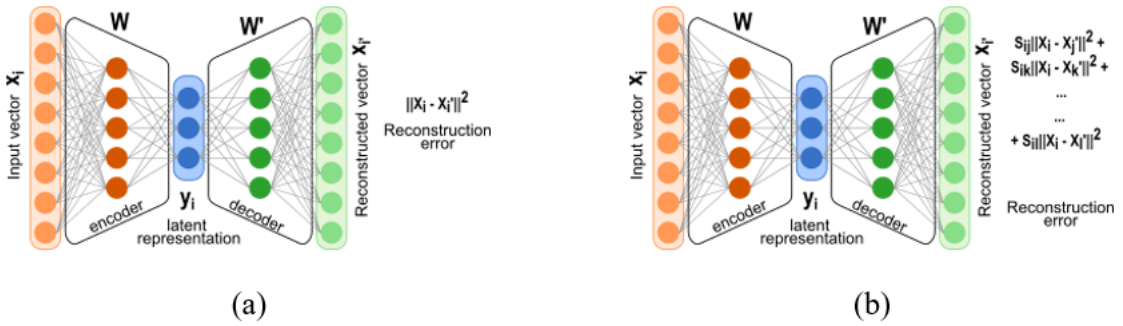

Fig 1. (a) Architecture of autoencoder and (b) architecture of generalized autoencoder. The difference between both architectures is that the autoencoder uses distance between x_i and x_i' i.e. $\left\|x_i - x_{i'}\right\|^2$ as reconstruction error, whereas generalized autoencoder uses weighted distance between x_i and x_j' i.e. $s_{ij}\left\|x_i - x_{j'}\right\|^2$

2.2.2. Generalized Autoencoder (GAE):

GAE[11] is a generalized framework for finding the lower-dimensional representation of data by manifold learning. Figure 1(b) shows the architecture of GAE. In this type of autoencoder the compressed representation is learned by building relationship between a data point x_i and other data points $\{\mathbf{x}_j, \mathbf{x}_k, ..., \mathbf{x}_l\}$ which are similar to $\mathbf{x}_i$.

Let Ω_i = *{j, k, ...}* be the reconstruction set for data point $\mathbf{x}_i$ that contains indices of other data points along with corresponding reconstruction weight set $S_i = \{s_{ij}, s_{ik}, ...\}$. GAE takes the reconstruction error between a data point x_i and data points from corresponding reconstruction set Ω_i. GAE penalizes this reconstruction error $\mathbf{x}_i$ and $\mathbf{x}_i$' by weight s_{ij}, which is a function of their pairwise similarity.

In GAE, hyper-parameters $(\mathbf{W}, \mathbf{W}^T)$ are trained by minimizing the weighted sum of squared reconstruction error given as:

$$E(\mathbf{W}, \mathbf{W}^T) = \sum_{i=1}^{M} \sum_{j \in \Omega_i} S_{ij} L(\mathbf{x_i}, \mathbf{x_j}') \tag{7}$$

With this formulation of GAE, selecting appropriate reconstruction set and reconstruction weight, dimensionality reduction similar to that of PCA, LDA, MFA, ISOMAP, LLE and LE can be implemented. Only GAE-LE and GAE-LLE are discussed here.

GAE-LE:

To implement GAE-LE, reconstruction set is set to $\Omega_i = N_i$ and reconstruction weight S_{ij} is calculated by:

$$S_{ij} = \begin{cases} \exp\left(-\dfrac{\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2}{t}\right), & j \in N_i \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

Here, *t*is a tuning parameter. The performance of GAE-LE highly depends on the choice of *t*.

GAE-LLE*:*

To implement GAE-LLE, reconstruction set is set to $\Omega_i = N_i$ and reconstruction weight is calculated by solving **Eq. 3**. The closed form solution is:

$$\mathbf{w_{i:}} = \frac{\mathbf{G_i}^{-1}\mathbf{e}}{\mathbf{e}^T\mathbf{G_i}^{-1}\mathbf{e}} \tag{9}$$

Here, $\mathbf{w}_{i:}$ is the i^{th} column in local reconstruction matrix $\mathbf{W}$, $\mathbf{e} \in \mathbb{R}^k$ is a vector of ones and $\mathbf{G}_i$ is a $k \times k$ matrix, also known as Gramian matrix for data point $\mathbf{x}_i$. Each entry of $\mathbf{G}$ is calculated as $G_{pq} = \left(\mathbf{x}_p - \mathbf{x}_q\right)^T \left(\mathbf{x}_p - \mathbf{x}_q\right), \forall \mathbf{x}_p, \mathbf{x}_q \in N_i$. The reconstruction weights for LLE can then be computed as:

$$S_{ij} = (\mathbf{W} + \mathbf{W}^T - \mathbf{W}^T\mathbf{W});\ if\ i \neq j \tag{10}$$

Table 1. GAE algorithm

Input	$\mathbf{X} \in \mathbb{R}^{m \times M}$
Output	$\mathbf{Y} \in \mathbb{R}^{d \times M}$
Hyper-parameter	**W :**encoder weights**,** $\mathbf{W^T}$ **:**decoder weights
Notations	Ω_i: Reconstruction set for $\mathbf{x}_i$ S_i: Reconstruction weight for $\mathbf{x}_i$
Steps	• Initialize $\Theta\left(\mathbf{W}, \mathbf{W}^T\right)$ • Compute the reconstruction set Ω_i and reconstruction weight S_i as in Table 2. • Minimize the cost function in **Eq. 6** using stochastic gradient descent and update Θ • Compute the hidden representation **Y** • Update Similarity weight set S_i and Reconstruction set Ω_i from **Y** • Repeat steps 3 to 5 until convergence

2.3 Proposed Modified GAE (MGAE)

LE uses heat kernel (**Eq. 8**) to assign reconstruction weights based on the Euclidean distance, also the effectiveness of LE depends on the choice of tuning parameter *t*. On the other hand, LLE also builds a linear relationship in a localized neighbourhood decided based on Euclidean distance from each data point and then extends this linearity assumption to the lower dimension space. However, this linearity assumption may not hold true in some applications, especially for data like face images, handwritten texts etc. where manifolds are proved to be highly nonlinear.

To overcome the limitations of rigid, rule-based weighting, a nonlinear weighting scheme is introduced. This approach employs a piecewise nonlinear function within local neighborhoods, resulting in a more compact data representation suitable for recognition tasks. Studies have shown that this type of weighting enhances recognition performance in image data [12].

In the proposed modification, a Z -function is employed to compute the reconstruction weights of the nearest neighbours. In LE, the penalty matrix P is derived using a heat kernel, which assigns weights inversely proportional to the distance between data points. Similarly, in Locally Linear Embedding (LLE), the weight matrix W is obtained by minimizing the cost function in **Eq. 2**, assigning weights inversely proportional to the distance between the point of interest and its neighbours.

The introduction of the Z -function creates a framework where neighbors closest to the point of interest receive the highest weights, which then decrease non-linearly as the distance increases. Beyond a certain threshold, the weights become negligible. This transition from high to low weights is controlled by a hyperparameter within the Z -function, enabling a flexible, piecewise relationship. This ultimately leads to a non-linear weighting scheme described by:

$$Z(d_{ij};a,b) = \begin{cases} 1 & \text{if } d_{ij} \le a \\ 1-2\left(\dfrac{d_{ij}-a}{a-b}\right)^2 & \text{if } a \le d_{ij} \le \dfrac{a+b}{2} \\ 2\left(\dfrac{d_{ij}-b}{a-b}\right)^2 & \text{if } \dfrac{a+b}{2} \le d_{ij} \le b \\ 0 & \text{Otherwise} \end{cases} \tag{11}$$

Here, d_{ij} is distance between data point $\mathbf{x}_i$ and $\mathbf{x}_j$ i.e. $\left\|\mathrm{x_i} - \mathrm{x_j}\right\|^2$. When using weighing scheme given in **Eq. 11**, parameter *a* and *b* are experimentally set based on the data. In our experiments, it is seen that *a* = 0 and *b* = data diameter (i.e. maximum pairwise distance between data points) works well and gives best results.

Similar to GAE-LLE, the reconstruction weight for each neighbour x_j corresponding to x_i is assigned using **Eq. 12**. It is similar to **Eq. 9**, where $\mathbf{G}^{-1}$ is replaced by **Z**. The new weights are

$$w_{i:} = \frac{Ze}{e^T Ze} \tag{12}$$

where, each element of **Z** matrix is defined as

$$\mathbf{Z}_{pq} = Z(d_{ip};a,b) + Z(d_{iq};a,b) for,\ \forall \mathbf{x_p}, \mathbf{x_q} \in N_i \tag{13}$$

Here, (d_{ip}; *a, b*) is calculated using **Eq 10**. d_{ip} is the Euclidean distance between $\mathbf{x}_i$ and its neighbor $\mathbf{x}_p$. Parameters *a* and *b* are set as discussed earlier.

2.4 Proposed Supervised GAE (SGAE)

GAE only considers *k*-nearest neighbours as reconstruction set of a data point $\mathbf{x}_i$, regardless of their class label. It is proved that using class knowledge in manifold learning not only incorporates data relationships but also incorporates discriminating information and yields improved classification performance. Taking this into account, we propose a method to incorporate class labels in GAE training named Supervised GAE (SGAE). In this Supervised GAE, the reconstruction set is decided based on the class label of data points.

Let c_i and c_j be the class labels of data point $\mathbf{x}_i$ and $\mathbf{x}_j$, respectively. If $c_i = c_j$, only then $x_j \in \Omega_{N_i}$. In other words, only data points belonging to the same class will be considered in each other's reconstruction set regardless of their Euclidean distance. Let Ω_{c_i} be the index set of the class $c_{i,}$ which $\mathbf{x}_i$ belongs to. Thus, $\mathbf{x}_j$ is in the reconstruction set if j $\in$ Ω_{c_i} . Note that in this case, we need not set the number of nearest neighbours *k* explicitly. Reconstruction weights will be calculated in the same manner as in Modified GAE.

Any of the proposed GAE implementations - MGAE-LE, MGAE-LLE, SGAE-LE, or SGAE-LLE - can be applied using their respective reconstruction sets and weights, following the algorithm outlined in Table 1. The reconstruction sets and weights are summarized in Table 2.

Table 2. Six different implementations of GAE, Modified GAE and Supervised GAE

Method	Reconstruction set	Reconstruction weight
GAE-LE	$\mathbf{x_j} \in N_i$	$\mathbf{s_{ij}} = \exp\left(\left\|\mathbf{x_i}-\mathbf{x_j}\right\|^2/\mathbf{t}\right)$
MGAE-LE	$\mathbf{x_j} \in N_i$	$\mathbf{s_{ij}} = \mathbf{Z}\left(\left\|\mathbf{x_i}-\mathbf{x_j}\right\|^2\right)$
SGAE-LE	$j \in \Omega_{c_i}$	$\mathbf{s_{ij}} = \mathbf{Z}\left(\left\|\mathbf{x_i}-\mathbf{x_j}\right\|^2\right)$
GAE-LLE	$\mathbf{x_j} \in N_i, \quad i \neq j$	$s_{ij} = \begin{cases} \mathbf{W}+\mathbf{W}^T-\mathbf{W}^T\mathbf{W} & if\ i \neq j \\ 0 & otherwise \end{cases}$
MGAE-LLE	$\mathbf{x_j} \in N_i, \quad i \neq j$	$s_{ij} = \begin{cases} \mathbf{Z}+\mathbf{Z}^T-\mathbf{Z}^T\mathbf{Z} & if\ i \neq j \\ 0 & otherwise \end{cases}$
SGAE-LLE	$j \in \Omega_{c_i}, \quad i \neq j$	$s_{ij} = \mathbf{Z}+\mathbf{Z}^T-\mathbf{Z}^T\mathbf{Z} \quad if\ i \neq j$

2.5 Experimental setup

Face and handwritten digit datasets are standard benchmarks in image recognition and give a reliable basis for comparing with existing methods. They also cover two different challenges —faces have complex variations in features, while handwritten digits show high variability within the same class. Therefore, we tested the proposed methods on eight benchmark datasets, including both face and handwritten digit images. The details of these datasets are given below.

2.5.1. Face datasets

The ORL dataset[13] has images from 40 individuals, 10 of each with variations in details, expressions, and poses as shown in Figure 2(a). The AR face database[14] contains nearly 4,000 images of 126 subjects captured in two different sessions two weeks apart, showing frontal face under different illumination, expressions, and occlusions like glasses and scarves (Figure 2(b)). The UMIST dataset[15] includes images of 20 people, between 19 to 36 poses of everyone from left to right profile (Figure 2(c)). The extended YALE-B dataset[16] has images of 28 individuals with 9 different poses under 64 illumination conditions (Figure 2(d)). All images were resized to 32 $\times$ 32 pixels to keep the input vector length uniform for all databases. For each dataset, 75% images were used for training and 25% for testing.

2.5.2. Digit datasets

For handwritten numeral recognition, four datasets were used: MNIST[17], Gujarati[18], Devnagari and Bangla[19]. MNIST contains approximately 68,000 images (Figure 3(a) shows samples of digit '2'). The Gujarati dataset includes around 13,000 images resized to 30 $\times$ 30 pixels, with samples of digit '7' shown in Figure 3(b). The Devnagari dataset has about 15,000 images, with digit '4' samples in Figure 3(c). The Bangla dataset contains nearly 20,000 images, with digit '8' samples in Figure 3(d). To maintain uniformity, all images were resized to 30 $\times$ 30 pixels. In each dataset, around 75% and 25% of the images were used for training and validation, respectively.

Fig 2. Example Images of one person from each of the benchmark Face Databases (a) ORL (b) AR (C) UMIST (D) Yale-B

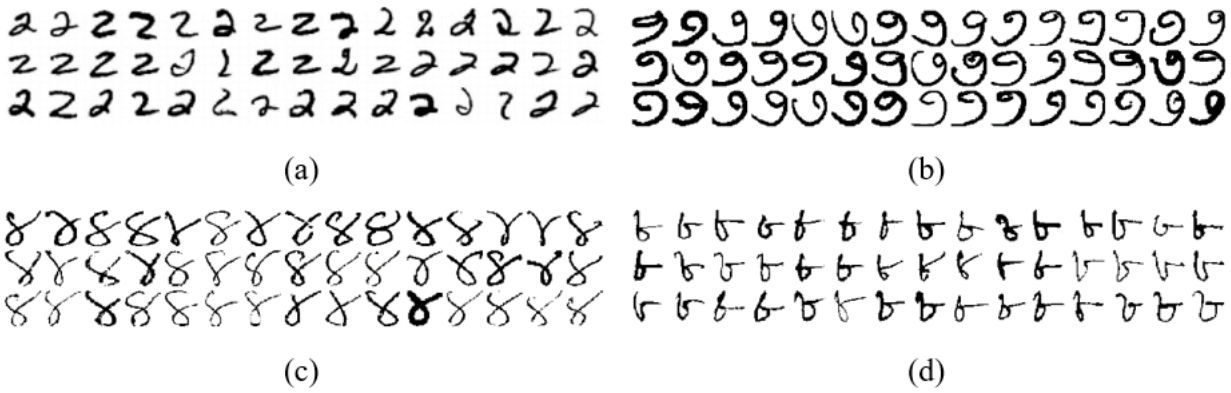

Fig 3. Example images from handwritten numeral datasets: (a) MNIST, (b) Gujarati, (c) Devanagari, and (d) Bangla

The GAE architecture used here is id $\rightarrow$ 300 $\rightarrow$ 150 $\rightarrow$ l_l $\rightarrow$ 150 $\rightarrow$ 300 $\rightarrow$ id. Here, id is input image dimensions and L_l is latent dimension. For face images l_l is set to 50; for digit images l_l is set to 30. The network is fully connected, with ReLU activation used for all hidden layers except the central l_l -dimensional bottleneck layer, which uses a linear activation to preserve the embedding structure. The model is trained using mean squared error (MSE) loss to minimize reconstruction error, and Adam optimizer with a learning rate of 0.001 is used for optimization. Training is performed for 500 epochs with a batch size of 128, and early stopping is applied based on validation loss to prevent overfitting. Input images are normalized to [0,1], and no data augmentation is applied in these experiments. Nearest Neighbour classifier is used on latent vectors for classification.

3 Results and Discussion

3.1 Face Image Recognition

F**ig.4** shows the comparison between recognition accuracy achieved using conventional LPP[20], GAE-LE[11], MGAE- LE (Z -weight) and SGAE-LE (Z -weight). As LE is a non-linear DR method, thus out-of-sample embedding is not possible with LE. Instead, its linear variant LPP is used for comparison. Figure 5 shows the comparison between recognition accuracy achieved

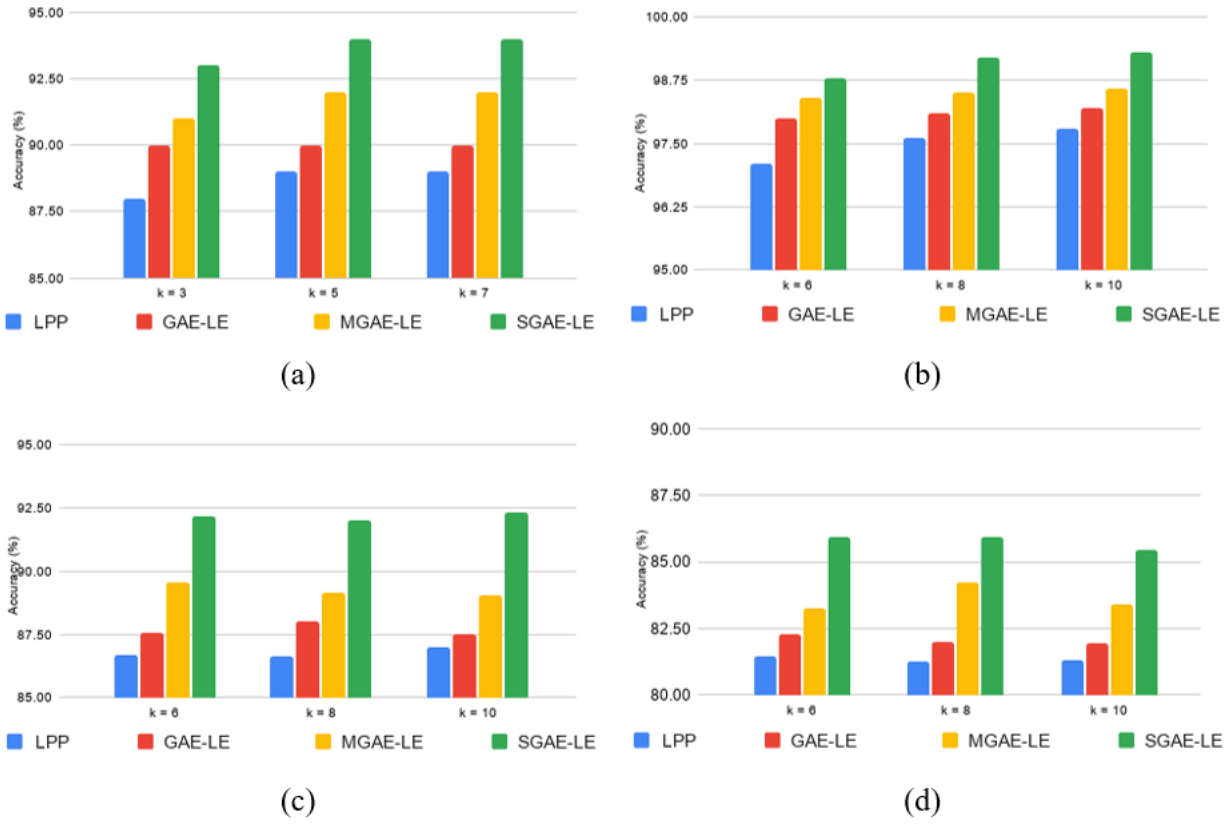

Fig 4. Recognition accuracy achieved for the respective Face databases using Proposed Modified and Supervised GAE-LE and comparison with base algorithms LPP, GAE-LE. Latent dimensions: 50, Classifier: NN (a) ORL (b) AR (C) UMIST (D) Yale-B

with GAE-LLE and its variants. To handle 'out-of-sample' problem, in place of LLE, its linear variant NPP[21] is used for comparison.

Table 3. Comparison of proposed methods against their counterpart existing methods on four benchmark face databases

	Existing Methods		**Proposed methods**		**Approximate Accuracy Gain**	
	LPP[20]	**GAE-LE[11]**	**MGAE-LE**	**SGAE-LE**	**modified weight**	**supervised neighbours**
ORL	88.02 ± 1.34	89.25 ± 0.16	92.19 ± 1.21	94.08 ± 1.24	4 %	6 %
AR	96.87 ± 0.74	97.43 ± 0.83	98.57 ± 0.89	98.73 ± 1.03	1 %	2 %
UMIST	86.98 ± 0.68	87.21 ± 0.71	89.06 ± 0.69	92.21 ± 0.58	2 %	5 %
Yale-B	81.35 ± 0.16	82.17 ± 0.73	83.98 ± 1.29	86.04 ± 0.63	2 %	5 %
	NPP[21]	**GAE-LLE[11]**	**MGAE-LLE**	**SGAE-LLE**		
ORL	89.04 ± 0.24	90.87 ± 0.53	91.13 ± 0.75	94.17 ± 0.84	2 %	5 %
AR	98.17 ± 0.76	98.72 ± 0.47	99.01 ± 0.50	99.71 ± 0.65	1 %	2 %
UMIST	87.13 ± 0.92	87.45 ± 0.76	89.23 ± 0.28	92.85 ± 0.92	2 %	6 %
Yale-B	81.78 ±0.63	82.03 ± 0.34	84.02 ±0.26	86.27 ± 0.93	3%	5%

Table 3 reports the average and standard deviation achieved in 20 randomized recognition experiments using four existing methods[11],[20],[21] and four proposed algorithms on face databases. The results highlight that use of Z -shaped weighing function increases recognition accuracy by 1-4% in MGAE-LE and 1-3% in MGAE-LLE, while introducing supervised neighbourhood selection increases recognition accuracy by 2-6% in both SGAE-LE and SGAE-LLE compared to LPP[20],

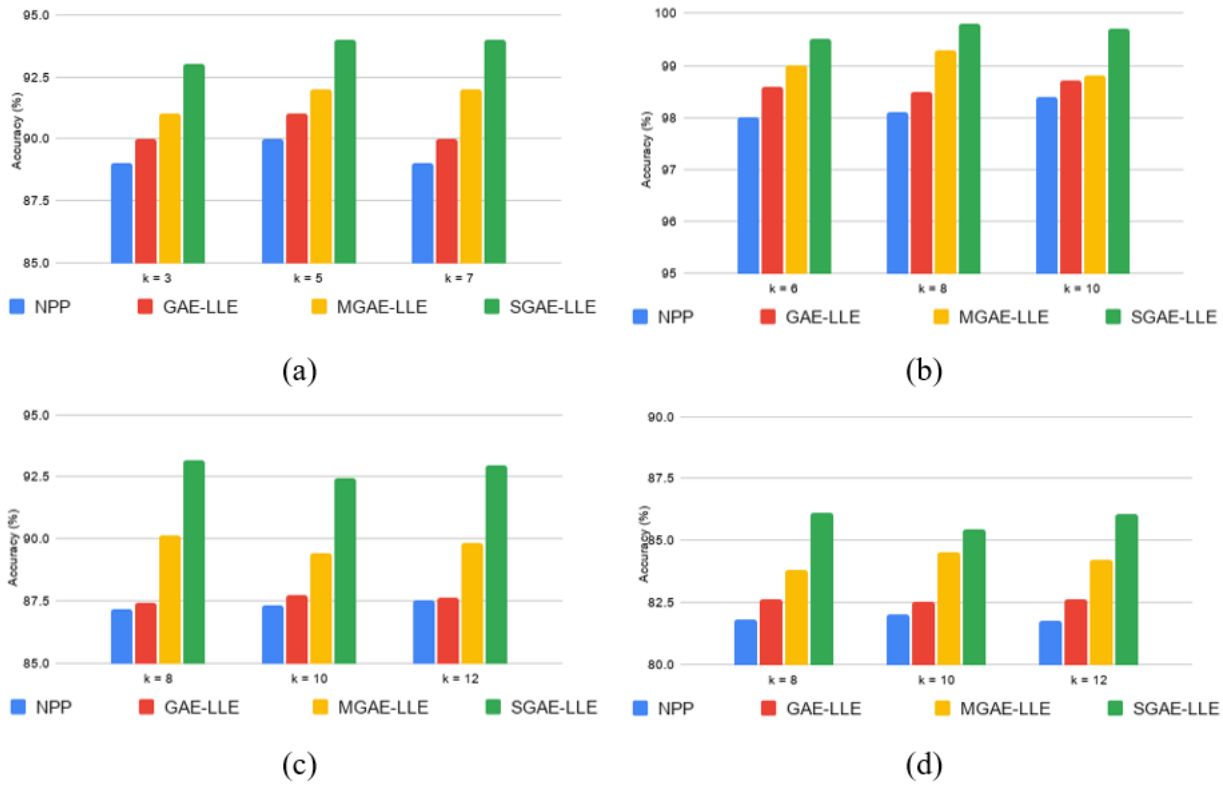

Fig 5. Recognition accuracy achieved for the respective Face databases using Proposed Modified and Supervised GAE-LLE and comparison with base algorithms NPP, GAE-LLE. Latent dimensions: 50, Classifier: NN (a) ORL (b) AR (C) UMIST (D) Yale-B

NPP[21] and GAE-LE,GAE-LLE[11]. As documented[12], the variants of LLE are outperforming variants of LE in recognition tasks. Similar behavior is observed in our experiments with MGAE-LE - MGAE-LLE and SGAE-LE – SGAE-LLE.

3.2 Digit Image Recognition

Figure 6 shows the comparison between recognition accuracy achieved using conventional LPP[20], GAE-LE[11], MGAE-LE (Z -weight) and SGAE-LE (Z-weight). Figure 7 shows the comparison between recognition accuracy achieved with GAE-LLE and its variant, in place of LLE, its linear variant NPP[21] is used for comparison.

Table 4. Comparison of proposed methods against their counterpart existing methods on four handwritten digits databases

	Existing Methods		**Proposed methods**		**Approx. Accuracy Gain**	
	LPP[20]	**GAE-LE[11]**	**MGAE-LE**	**SGAE-LE**	**modified weight**	**supervised neighbours**
MNIST	91.74 ± 0.54	92.34 ± 0.67	94.81 ± 0.29	95.28 ± 0.88	3 %	4 %
Gujarati	89.16 ± 0.91	91.87 ± 0.46	92.41 ± 0.73	96.31 ± 0.32	3 %	6 %
Devnagari	89.27 ± 0.58	89.71 ± 0.85	92.07 ± 0.41	94.82 ± 0.93	3 %	5 %
Bangla	87.82 ± 0.26	89.86 ± 0.63	91.03 ± 0.77	94.81 ± 0.38	3 %	7%
	NPP[21]	**GAE-LLE[11]**	**MGAE-LLE**	**SGAE-LLE**		
MNIST	92.13 ± 0.72	92.89 ± 0.35	95.19 ± 0.89	96.06 ± 0.44	3 %	4 %
Gujarati	90.25 ± 0.56	91.92 ± 0.27	92.33 ± 0.68	96.98 ± 0.92	2 %	7 %
Devnagari	88.71 ± 0.33	90.27 ± 0.74	91.67 ± 0.48	95.31 ± 0.61	3 %	6 %
Bangla	87.94 ± 0.86	90.23 ± 0.29	91.77 ± 0.53	95.03 ± 0.81	4 %	7 %

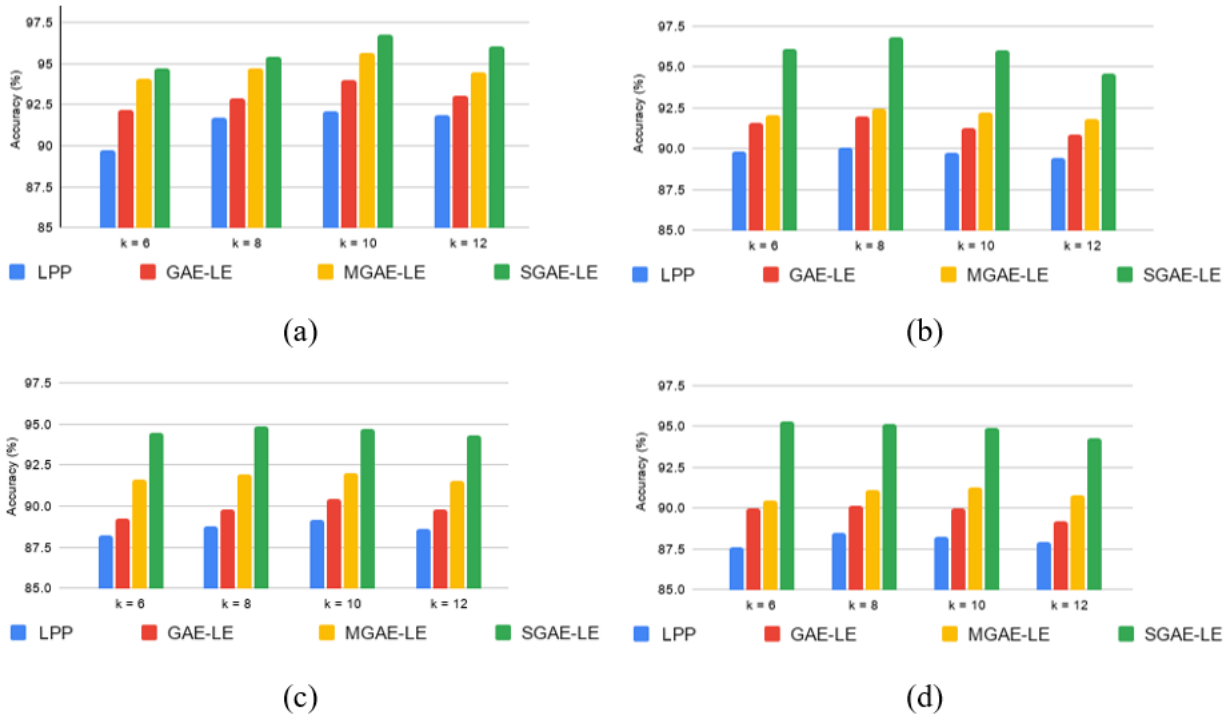

Fig 6. Comparison of recognition accuracy achieved for Handwritten Numerals databases using Proposed Modified and Supervised GAE-LE with base algorithm LPP and GAE-LE. Latent dimensions: 30 (a) MNIST (b) Gujarati (c) Devnagari (d) Bangla

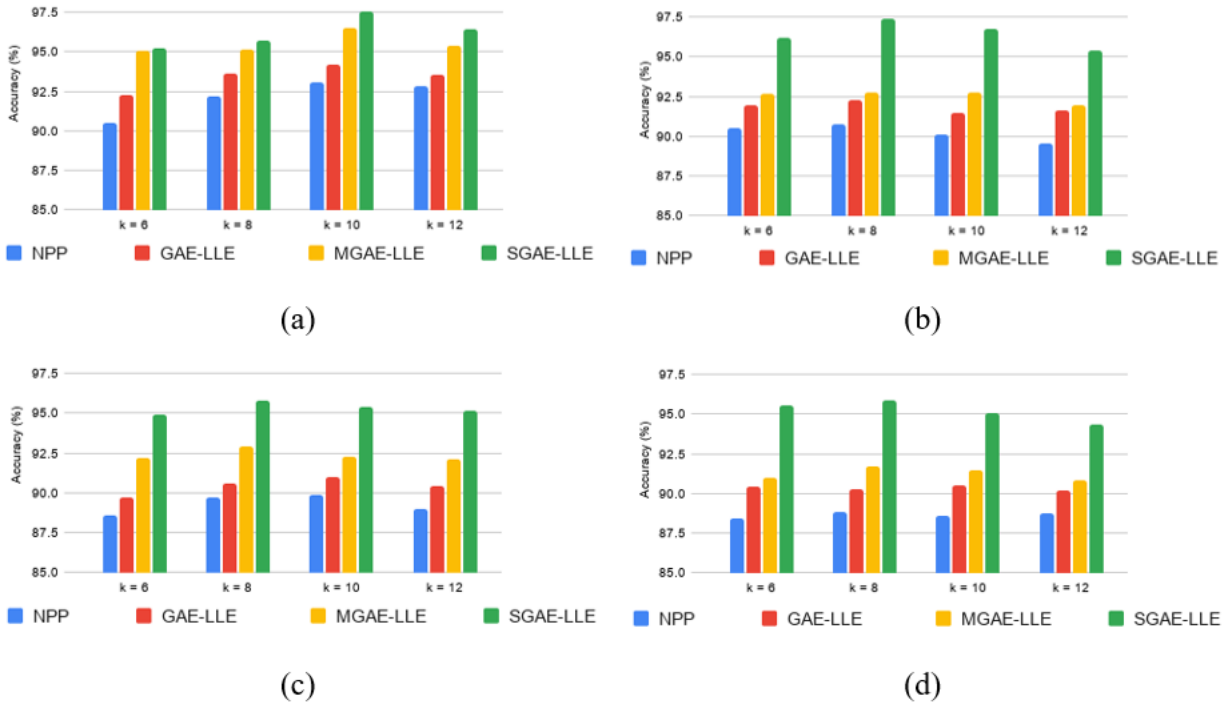

Fig 7. Comparison of recognition accuracy achieved for Handwritten Numerals databases using Proposed Modified and Supervised GAE-LLE with base algorithms NPP and GAE-LLE. Latent dimensions: 30 (a) MNIST (b) Gujarati (c) Devnagari (d) Bangla

Table 4 summarizes the average and standard deviation achieved in 20 randomized recognition experiments using four existing methods[11],[20],[21] and four proposed algorithms on digits databases. Use of Z -shaped weighing function increases recognition accuracy by approx. 3% in MGAE-LE and 2-4% in MGAE-LLE, while introducing supervised neighbourhood selection increases recognition accuracy by approximate 4-7% in both SGAE-LE and SGAE-LLE compared to LPP[20], NPP[21] and GAE-LE,GAE-LLE[11]. The effectiveness of supervised modification is evident from the fact that when limited number of labelled data is available, like in Bangla, the proposed method brings major improvement (7%) in recognition accuracy.

3.3 Comparison with existing literature and novelty

The improvements reported in Tables 3– 4 show that MGAE and SGAE consistently outperform classical linear variants (LPP/NPP) and the previously proposed GAE variants under the same experimental setup. Compared to prior manifold-aware autoencoder and Laplacian-regularized AE studies[2],[3],[4], our methods differ in two key ways: (i) MGAE introduces a non-linear, shape-controlled weighting function that produces more accurate local reconstructions and thus a more discriminative embedding, and (ii) SGAE explicitly injects supervised neighbour selection into the reconstruction set so that class structure guides embedding formation when labels are available. Quantitatively, MGAE yields ≈1–4% absolute gains over conventional DR and GAE baselines across face datasets, while SGAE brings an additional ≈2–7% improvement, with the largest gains observed on datasets with limited labelled samples (e.g., Bangla and Gujarati). These results complement earlier reports where topology-preserving and Laplacian AEs improved representational quality but were largely unsupervised[2],[3] by demonstrating that modest supervision targeted at neighbour selection can yield substantial discriminative benefits without major architectural changes. Finally, while the accuracy gains are statistically significant (see test results in Section 3.2 and 3.3), they come at a moderate computational cost due to the additional weighting and supervised neighbours selection steps.

4 Conclusion

This work proposed two enhanced versions of the Generalized Autoencoder (GAE) to improve image recognition performance. The first, MGAE, introduces a novel non-linear weighing scheme that improves reconstruction accuracy, while the second, SGAE, incorporates class label information to enable better class discrimination. Experiments on four face datasets and four handwritten digit datasets show that MGAE improves accuracy by 1–4% over conventional DR methods and GAE, and SGAE provides an additional 2–7% improvement compared to their unsupervised counterparts.

The proposed methods demonstrate stronger performance and flexibility, especially in scenarios with limited labelled data. However, computational costs may increase due to the additional weighting and supervision. Further testing on semi-supervised approaches could enhance the applicability of these methods. Overall, MGAE and SGAE offer promising directions for improving dimensionality reduction and recognition accuracy in practical image analysis applications.

5 Author Contributions

All authors were equally involved in conceptualization, methodology design, data analysis, manuscript drafting, and review. Each author has read and approved the final version of the manuscript.

References

1) Berahmand K, Daneshfar F, Salehi ES, Li Y, Xu Y. Autoencoders and their applications in machine learning: a survey. *ArtifIntell Rev*. 2024;57:28. https://doi.org/10.1007/s10462-023-10662-6.
2) Zhao X, Jia M, Lin M. Deep Laplacian auto-encoder and its application into imbalanced fault diagnosis of rotating machinery. *Measurement*. 2020;152:107320. https://doi.org/10.1016/j.measurement.2019.107320.
3) Chen N, van der Smagt P, Cseke B. Fast preserving local distances and topology in auto-encoders. *Neural Information Processing ICONIP 2024*. 2025;15287. https://doi.org/10.1007/978-981-96-6579-2_28.
4) Paricio-Garcia A, Lopez-Carmona MA, Sierra-Arquero S, Manglano-Redondo P. Analysis and evaluation of autoencoder-driven dimensionality reduction for face recognition pipelines. *Appl Soft Comput*. 2025;172:112877. https://doi.org/10.1016/j.asoc.2025.112877.
5) Triveni G, Nayak SK, Padhy J, Gunalakshmi C, Sehgal M, Choudari S. Image denoising and dimensionality reduction using autoencoder. *2024 2nd International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*. 2024;p. 1–4. https://doi.org/10.1109/SCOPES64467.2024.10991331.
6) Ali I, Wassif K, Bayomi H. Dimensionality reduction for images of IoT using machine learning. *Sci Rep*. 2024;14:7205. https://doi.org/10.1038/s41598-024-57385-4.
7) Mienye ID, Swart TG. Deep autoencoder neural networks: a comprehensive review and new perspectives. *Arch Comput Methods Eng*. 2025. https://doi.org/10.1007/s11831-025-10260-5.
8) Belkin M, Niyogi P. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput*. 2003;15(6):1373–1396. https://doi.org/10.1162/089976603321780317.

9) Roweis ST, Saul LK. Nonlinear dimensionality reduction by locally linear embedding. *Science*. 2000;290(5500):2323–2326. https://doi.org/10.1126/science.290.5500.2323.
10) Wang Z, Zeng Q, Lin W, Jiang M, Tan KC. Multi-view subgraph neural networks: self-supervised learning with scarce labeled data. *IEEE Trans Neural Netw Learn Syst*. 2024;36(6):11548–11561. https://doi.org/10.1109/TNNLS.2024.3443074.
11) Wang W, Huang Y, Wang Y, Wang L. Generalized autoencoder: a neural network framework for dimensionality reduction. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2014;p. 490–497. https://doi.org/10.1109/CVPRW.2014.79.
12) Koringa P, Shikkenawis G, Mitra SK, Parulkar SK. Modified orthogonal neighborhood preserving projection for face recognition. Springer. 2015. https://doi.org/10.1007/978-3-319-19941-2_22.
13) Samaria FS, Harter A. Parameterisation of a stochastic model for human face identification. *2nd IEEE Workshop on Applications of Computer Vision*. 1994. Available from: https://git-disl.github.io/GTDLBench/datasets/att_face_dataset/. https://doi.org/10.1109/ACV.1994.341300.
14) Martinez A, Benavente R. The AR face database. Barcelona. Computer Vision Center. 1998. Available from: https://service.tib.eu/ldmservice/dataset/ar-face-database.
15) Graham DB, Allinson NM. Characterising virtual eigensignatures for general purpose face recognition. NATO ASI Series F. unknown. Available from: https://opendatalab.com/OpenDataLab/UMIST. https://doi.org/10.1109/34.927464.
16) Georghiades AS, Belhumeur PN, Kriegman DJ. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans Pattern Anal Mach Intell*. 2001;23(6):643–660. https://doi.org/10.1109/34.927464.
17) LeCun Y, Cortes C. The MNIST database of handwritten digits. 1998. Available from: http://yann.lecun.com/exdb/mnist/.
18) Nagar R, Mitra SK. Feature extraction based on stroke orientation estimation technique for handwritten numerals. *8th International Conference on Advances in Pattern Recognition (ICAPR)*. 2015;p. 1–6. https://doi.org/10.1109/ICAPR.2015.7050654.
19) Bhattacharya U, Chaudhuri BB. Databases for research on recognition of handwritten characters of Indian scripts. *8th International Conference on Document Analysis and Recognition (ICDAR)*. 2005;p. 789–793. https://doi.org/10.1109/ICDAR.2005.84.
20) He X, Niyogi P. Locality preserving projections. *Advances in Neural Information Processing Systems*. 2004;p. 153–160.
21) He X, Cai D, Yan S, Zhang HJ. Neighborhood preserving embedding. *Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV)*. 2005;2:1208–1213. https://doi.org/10.1109/ICCV.2005.167.