Shahid Bahonar University of Kerman
Journal of Mahani Mathematical Research
Print ISSN: 2251-7952 Online ISSN: 2645-4505

UNSUPERVISED FEATURE SELECTION USING ORTHOGONAL LOCALITY PRESERVING PROJECTIONS AND BIPARTITE GRAPH MATCHING FOR FACE IMAGE CLASSIFICATION

F. Beiranvand ✉

Article type: Research Article
(Received: 14 February 2025, Received in revised form 06 August 2025)
(Accepted: 17 September 2025, Published Online: 17 September 2025)

Abstract. Feature selection plays a crucial role in facial image classification by reducing dimensionality and improving robustness to variations in expression, pose, and lighting. However, researchers face challenges when selecting features from high-dimensional, unlabeled data due to the nonlinear manifold structure of facial images. To address this, this paper proposes UFSOLPP, a novel unsupervised feature selection method that consists of three main stages. First, the method employs Orthogonal Locality Preserving Projections (OLPP) for feature extraction, aiming to preserve local data structures and enforce orthogonality without dimensionality reduction. Unlike conventional OLPP, which uses heat kernel to measure similarity, this paper replaces it with cosine distance to better capture angular relationships that are for facial image discrimination. Second, it measures the similarity between the original and orthogonal features using the Pearson correlation distance. Third, it models both feature sets as vertices in a weighted bipartite graph. The edge weights are computed using the Pearson correlation similarity, and the method uses the Hungarian algorithm to compute maximum matching. The method selects the original features involved in the maximum matching as the final subset. This strategy removes noisy, correlated, and redundant features effectively, while preserving interpretability and discriminative power. Experiments demonstrate that UFSOLPP outperforms eight state-of-the-art methods. It achieves 96.00% accuracy and 98.00% NMI on Jaffe, 68.66% accuracy and 75.32% NMI on ORL, and 82.33% accuracy and 85.57% NMI on the high-dimensional Pixraw10P dataset. These results highlight the practical value of UFSOLPP and its ability to handle high-dimensional data efficiently in unsupervised facial image analysis.

Keywords: OLPP, manifold learning, weighted bipartite graph matching, Hungarian algorithm
2020 MSC: 68T20.

✉ beiranvand.fi@fe.lu.ac.ir, ORCID: 0000-0002-1908-8386
https://doi.org/10.22103/jmmr.2025.24829.1765
Publisher: Shahid Bahonar University of Kerman
How to cite: F. Beiranvand, *Unsupervised feature selection using orthogonal locality preserving projections and bipartite graph matching for face image classification*, J. Mahani Math. Res. 2026; 15(2): 65-104.

1. Introduction

High-dimensional data are common in a wide range of applications, including object recognition, image restoration, information retrieval, and computer vision. Although these datasets provide richer information, they also lead to substantial increases in computational and storage costs [1]. To address these challenges, dimensionality reduction has become a crucial preprocessing step in machine learning and data mining workflows. Dimensionality reduction techniques generally fall into two main categories: feature selection and feature extraction. Feature extraction methods aim to map high-dimensional data into a lower-dimensional space through algebraic transformations [2], and can be further classified into linear and nonlinear approaches. Linear methods assume that data lie in a linear subspace, whereas nonlinear methods are designed to capture complex structures on nonlinear manifolds [3]. Among nonlinear methods, manifold learning techniques are particularly notable, as they assume that data points reside on a low-dimensional manifold embedded within a higher-dimensional space. By preserving the intrinsic geometry of the data, manifold learning helps alleviate the curse of dimensionality [4].

However, while linear methods show promise, they often fail to adequately address the challenge of maintaining local structures in high-dimensional spaces, particularly when dealing with subtle variations in the data. For instance, in face image recognition, changes in facial expressions lead to nonlinear variations that are not easily captured by traditional linear methods. This limitation in existing techniques has motivated the development of more effective methods, specifically tailored to preserve these local structures and model the nonlinear variations effectively.

On the other hand, feature selection aims to eliminate redundant features, retaining only those most relevant to the target task. This process not only enhances the performance of classifiers but also helps reduce overfitting. Feature selection proves to be especially valuable across various fields, including industrial applications, bioinformatics, image processing, and text mining [5–8]. Feature selection methods are generally categorized into three main strategies: filter, wrapper, and embedded methods. Unlike wrapper and embedded methods, filter methods do not rely on learning algorithms. Instead, they assess feature relevance based on intrinsic properties of the data. Filter methods are particularly favored for high-dimensional datasets due to their computational efficiency. These methods can be further classified into univariate and multivariate approaches, with the latter accounting for correlations between features [9–11]. Wrapper methods, in contrast, determine feature importance by utilizing a classifier during the learning process [11]. Although wrapper methods are accurate, they are computationally expensive and susceptible to overfitting, especially when the number of features exceeds the number of samples [9]. Embedded methods incorporate feature selection directly into the

learning process by using cost functions that consider feature relevance, often involving sparse regularization [9, 13].

Moreover, feature selection can also be classified based on the availability of class labels into three categories: supervised, unsupervised, and semi-supervised [9]. In semi-supervised feature selection, some data points are labeled, and this label information is used to improve the performance of unsupervised feature selection methods [13]. In supervised feature selection, labeled data allow for the identification of relevant features based on their relationship with class labels. In contrast, unsupervised feature selection faces challenges due to the lack of labeled data, where feature relevance is inferred from intrinsic data characteristics, such as variance [14, 15].

In the task of feature selection for face images, the primary challenge stems from the inherent complexity of the data and the nonlinear variations in facial features, such as changes in facial expressions. These features are typically located on a nonlinear manifold, which complicates the identification of the most relevant and effective features. For example, variations in facial expressions, such as the transition from a neutral expression to a smile, involve complex changes in the curvature and shape of the lips, as well as the position of the eyebrows. In high-dimensional space, these variations form a curved manifold rather than a flat surface. Many existing methods struggle to accurately model these nonlinear manifolds and preserve the local structures of the data, which is essential for selecting meaningful features.

To address these challenges, recent approaches have focused on preserving local structures and reducing feature redundancy by leveraging pseudo-labels, adaptive graph learning, and hypergraph models [16]. However, in unsupervised feature selection, the absence of class labels remains a major challenge, especially when dealing with facial features that are geometrically and structurally similar but exhibit subtle variations, such as those caused by changes in facial expressions. To overcome these limitations, this paper proposes an unsupervised feature selection method based on the Orthogonal Locality Preserving Projections (OLPP) algorithm. This approach preserves the local structures of the data, facilitating the extraction of features that more effectively model the nonlinear variations in face images. The OLPP algorithm extracts orthogonal features, reduces feature correlations, thereby identifying the most distinct and informative features. Furthermore, OLPP mitigates the impact of noise and outliers by focusing on preserving local structures, filtering out irrelevant data points, and enhancing the quality of the extracted features. This ultimately leads to more accurate classification results, despite the inherent challenges posed by high-dimensional and nonlinear data.

The primary contributions of this paper are summarized as follows: 1) **OLPP for Feature Extraction:** This paper employs the Orthogonal Locality Preserving Projections (OLPP) method for feature extraction, aiming to project data while preserving local structures and enforcing orthogonality.

This approach enhances the representational quality of features and helps reduce the impact of noise and redundancy in the original set. It is important to note that, at this stage, the algorithm performs a transformation of the features into a new space without applying dimensionality reduction. In the conventional OLPP framework, the similarity between adjacent data points is typically calculated using the heat kernel. However, in the context of face image data, angular relationships between feature vectors often contain more discriminative information. To address this, the heat kernel is replaced with cosine distance, allowing the graph to better reflect angular separations. Although this modification improves the alignment with the geometric structure of facial features, it may result in a nonlinear and complex-valued matrix. To resolve this, the complex conjugate of the matrix is computed, followed by a linearization process.
2) **Feature Similarity Calculation:** A similarity calculation step is introduced using Pearson correlation to quantify the relationship between the original and extracted features. This step is essential for identifying the most relevant features for the classification task.
3) **Maximum Matching in Weighted Bipartite Graph:** The paper models the original and extracted features as vertices in a bipartite graph, using Pearson correlation as edge weights. By applying the Hungarian algorithm to find the maximum matching, the most discriminative features for classification are identified.
This method offers a robust framework for unsupervised feature selection, making it particularly effective for high-dimensional and nonlinear data, such as those commonly encountered in face image classification.

The primary objective of this research is to develop an unsupervised feature selection method that effectively addresses the challenges of high-dimensional and nonlinear data, particularly in face image recognition. Specifically, the proposed method aims to preserve the local structures of the data while mitigating the impact of noise and redundancy, and to identify the most discriminative features for classification tasks. Additionally, the research seeks to enhance the performance of unsupervised feature selection methods by employing the Orthogonal Locality Preserving Projections (OLPP) algorithm, which reduces feature dependencies and correlations, and utilizes the maximum matching in a weighted bipartite graph to further improve feature selection accuracy. This study contributes to the growing body of research by providing a more effective approach to feature selection in complex, high-dimensional datasets.

The structure of the paper is organized as follows:
Section 2 provides a brief review of recent advancements in feature selection research. Section 3 introduces and elaborates on the key concepts and theoretical framework underlying the proposed Unsupervised Feature Selection method based on Orthogonally Locality Preserving Projection (UFSOLPP). The experimental results are presented and discussed in Section 4, and Section

5 concludes the paper with a summary of the findings.

2. Related work

Numerous unsupervised feature selection algorithms have been developed to address the challenges of high-dimensional data. These methods differ in their assumptions, objectives, and mechanisms, including reconstruction, graph learning, redundancy reduction, and manifold preservation.

Wang et al. (2024) proposed MUNFS, which leverages UMAP for dimensionality reduction based on topological data analysis. By considering both local and global structures, MUNFS generates high-quality embeddings and integrates the block HSIC Lasso method to capture nonlinear relationships between input features and their low-dimensional representations [17]. Li et al. (2024) introduced DLSEO, which enhances clustering by using an extended OLSDA model to capture non-negative manifold structures and clustering labels. Its novelty lies in applying dual-space regularization and replacing the conventional $\ell_{2,1}$-norm with an $\ell_{2,1}$-2-norm for greater sparsity and lower feature redundancy [18]. Xiang et al. (2024) proposed BGLR, a bipartite graph-based model that adaptively selects anchors in the projection space using sample variance and correlation. This method combines an $\ell_{2,0}$-norm constraint with a low-redundant regularizer, allowing direct optimization of feature subsets [19]. Han et al. (2018) introduced AEFS, which assigns weights to features using a self-reconstructing autoencoder, enabling nonlinear modeling of feature importance. Although effective, AEFS depends heavily on hyperparameter tuning [20]. Guo et al. (2018) presented DGUFS, which jointly selects features and performs clustering by aligning feature selection with cluster label inference, enhancing the interdependence among data components. Its iterative optimization framework relies on ADMM and is sensitive to initialization and parameter settings [21]. Xie et al. (2021) proposed SCEFS and SCAFS, two methods that evaluate feature discriminability using standard deviation combined with exponential and anti-cosine similarity measures, respectively. These methods do not require labels but depend on empirical thresholding [22]. Zhu et al. (2015) introduced RSR, a regularized self-representation model that expresses each feature as a linear combination of others using an $\ell_{2,1}$-norm regularizer. Although robust to outliers, its effectiveness is influenced by regularization parameters [23]. Huang et al. (2019) proposed SRCFS, which applies randomized subspace learning and collaborative Laplacian scoring to enhance selection accuracy. The method depends on the number and size of subspaces, which must be carefully tuned [24]. Li et al. (2018) developed URAFS, a method that integrates manifold learning and feature selection through uncorrelated regression and entropy-based regularization. While effective, it requires multiple regularization terms and balancing parameters [25]. More recently, Samareh-Jahani et al. (2023) proposed LRDOR, which utilizes QR factorization to construct

orthogonal representations and applies a directional distance metric to evaluate feature relevance. The model incorporates local feature correlations and manifold structure, demonstrating competitive clustering performance. However, it lacks a direct matching mechanism between original and transformed features [26].

In another recent work, Xiao et al. (2025) introduced the JSLSC model, which combines subspace learning with orthogonal basis clustering and hard-constrained graph structure learning. This model excels in preserving both local and global structures, but it suffers from high model complexity and tuning requirements and is mainly designed for clustering rather than direct feature matching [27]. Beiranvand et al. (2022) proposed UFSPCA, which uses PCA to produce orthogonal features and applies the Hungarian algorithm on a bipartite graph for optimal feature matching. Although it reduces redundancy, it lacks the ability to preserve local structures in the data, limiting its effectiveness in complex setting. It is important to note that PCA, relies on the mean and variance of the data. This dependence makes PCA sensitive to noise and outliers, as they can significantly affect variance estimation and thus degrade feature selection accuracy [9]. Ma et al. (2024) developed SWAGFS, which jointly optimizes adaptive self-weighted graph construction, $\ell_{2,1}$-norm sparsity, and minimum redundancy constraints. This method adaptively learns feature importance by preserving the intrinsic geometric structure of the data. Its main novelty lies in integrating self-weighted graph learning and sparsity within a unified framework. However, its performance relies on sensitive hyperparameter tuning [28]. Zhang et al. (2024) proposed a supervised feature selection method that incorporates a trace ratio model with multi-center representation and local structure learning. By applying an $\ell_{2,0}$-norm constraint and utilizing k-nearest neighbor graphs, their method selects highly discriminative features. Although effective in labeled settings, it is not applicable to unsupervised scenarios due to its reliance on class labels [29]. Zohrati et al. (2018) introduced ahybrid max–min and greedy scheduling algorithm to optimize job allocation in cloud computing. The algorithm applies constraint-aware matching strategies in high-dimensional assignments [30]. A detailed comparison of the methods is provided in Figure 1.

While existing methods offer various strategies for feature selection, they often suffer from limitations such as sensitivity to parameters, lack of geometric preservation, or high computational costs. The proposed UFSOLPP method uses Orthogonal Locality Preserving Projections (OLPP) to extract orthogonal features while maintaining the local geometric structure of the data. It provides an intuitive graph-based representation of features that effectively captures their relationships. The method constructs a weighted bipartite graph between the original and projected features and applies the Hungarian algorithm to find the maximum matching. This process selects relevant and non-redundant features optimally. By doing so, the method preserves local structure, eliminates noise, and reduces feature redundancy. Moreover, UFSOLPP avoids the need

for extensive hyperparameter tuning and offers robustness and enhanced discriminative power in complex, high-dimensional unsupervised learning tasks.

Algorithm	Year	Key Idea	Advantages	Disadvantages	Data Structure	Parameter Dependency
UFSPCA [9]	2022	Uses PCA to generate orthogonal features and bipartite graph matching with the Hungarian algorithm	Reduces redundant features, simple implementation	Does not preserve local data structure, reduced discriminative power	Linear	Parameter-free
AEFS [20]	2018	Uses an autoencoder and assigns weights for each feature with nonlinear reconstruction	Models both linear and nonlinear dependencies	Requires tuning multiple hyperparameters, less stable	Nonlinear	Hyperparameter sensitive
DGUFS [21]	2018	Joint optimization of feature selection and clustering with ADMM	Enhances relationships between data, labels, and features	Sensitive to initialization and parameters	Linear	Parameter sensitive
SCEFS [22]	2021	Feature selection based on standard deviation and exponential cosine similarity	No label required, suitable for genomic data	Requires empirical threshold setting	Linear	Requires an empirical threshold
SCAFS [22]	2021	Feature selection based on standard deviation and anti-cosine similarity	Similar to SCEFS with a different similarity definition	Similar limitations as SCEFS	Linear	Requires an empirical threshold
RSR [23]	2015	Representation of each feature as a linear combination of others with ℓ2,1-norm	Robust to noise and outliers	Requires careful tuning of regularization parameters	Linear	Parameter sensitive
SRCFS [24]	2019	Uses random subspaces and collaborative Laplacian scoring	High selection accuracy	Depends on subspace size and partitioning setup	Nonlinear	Complex configuration
URAFS [25]	2018	Uncorrelated regression with maximum entropy-based regularization	Joint manifold learning and feature selection	Depends on multiple regularization parameters	Linear	Parameter sensitive
UFSOLPP (Proposed)	2025	Uses OLPP for orthogonal locality preserving projection, Pearson correlation, and bipartite graph maximum matching	Preserves local structure, orthogonality, parameter-light, robust	Computational cost of graph matching	Linear/Nonlinear	Low parameter sensitivity

FIGURE 1. Comparison of related unsupervised feature selection algorithms

3. Methodology

This section provides a comprehensive explanation of the proposed UFSOLPP method. Unsupervised feature selection methods often overlook critical features that significantly affect model performance, particularly in the presence of noise and outliers. This can lead to incorrect pattern recognition and unreliable models. In face image analysis, the complexity of the data and nonlinear variations, such as changes in facial expressions, pose significant challenges. These variations typically lie on nonlinear manifolds, making it difficult to identify discriminative features. For example, transitioning from a neutral expression to a smile involves subtle geometric changes in facial structure, which

are not effectively captured by linear methods. Consequently, many existing algorithms fail to model such nonlinear manifolds or preserve local data structures, resulting in suboptimal feature selection. Furthermore, the absence of labeled data in unsupervised feature selection complicates the task even more, especially when faces are structurally similar but differ in subtle ways.

To address these limitations, this paper proposes an unsupervised feature selection method called UFSOLPP, which preserves local data structures to effectively model nonlinear variations. The core idea is to represent features as vertices in a weighted bipartite graph, where edge weights indicate the similarity between features. By applying the Hungarian algorithm, the method identifies a maximum matching that selects features most similar to the orthogonal ones extracted using OLPP. This approach ensures the selection of informative and diverse features by reducing redundancy and correlation. The proposed method consists of three main steps:

1. Feature Extraction with OLPP: In the first step, the Orthogonal Locality Preserving Projection (OLPP) [31] method is employed for nonlinear feature extraction. OLPP preserves the local structure of data by maintaining the closeness of similar data points after projection while enforcing orthogonality to ensure feature uncorrelations [32]. This technique, rooted in manifold learning, formulates the embedding using the Laplacian graph structure, leading to effective nonlinear dimensionality reduction. Manifold learning performs nonlinear dimensionality reduction by utilizing the manifold structure among the samples. Its main idea is to apply the fundamental properties of the Laplacian graph to construct an optimal low-dimensional embedding of high-dimensional data, formulated as the solution to a generalized eigenvector problem [33]. Notably, in this step, OLPP is used without performing dimensionality reduction; instead, it extracts a new set of features with the same dimensionality as the original features.

2.Similarity Calculation: In the second step, the Pearson correlation similarity between the extracted features and the original features is calculated.

3.Feature Selection via Bipartite Graph Matching: In the third step, a weighted bipartite graph is constructed using the two feature sets, where the similarity scores serve as edge weights connecting the two parts of the graph. The best features are then selected by finding the maximum matching in the graph using the Hungarian algorithm. Figure 2 illustrates the flowchart of the proposed UFSOLPP method, and a detailed description of each step is provided in the following subsections.

3.1. **Orthogonal Locality Preserving Projection (OLPP).** The first step of the proposed UFSOLPP algorithm involves feature extraction using the fundamental OLPP algorithm, which is derived from the nonlinear Laplacian–based feature extraction method. OLPP is distinguished by its ability

to extract features that preserve orthogonality, non-correlation, and the locality of the original feature structure within the transformed space. Enhanced preservation of locality is directly associated with improved discriminative capability. In the following, the OLPP approach is explained step by step.

Step 1: The process begins by applying the Principal Component Analysis (PCA) method [9] to the training data X in order to extract orthogonal features. The eigenvectors obtained from the eigendecomposition of the PCA covariance matrix are stored in the matrix P_{PCA}. At this stage, no features are discarded.

Step 2: An adjacency graph G is constructed for the training data, where each of the n vertices represents a feature. An edge is formed between vertices i and j if feature x_{i} is among the k-nearest neighbors of x_{j}, and vice versa. In this implementation, $k = 5$. This graph captures the local geometric relationships among features based on the Euclidean distance [31]. The resulting structure, referred to as the local intrinsic graph S, serves as a discrete approximation of a low-dimensional manifold within the high-dimensional feature space. If features x_{i} and x_{j} lie on the same manifold, their output distributions z_{i} and z_{j} are expected to be similar [34]. This ensures that nearby points in the high-dimensional space remain close in the reduced space, preserving the local structure [35]. An example of a Local intrinsic graph is illustrated in Figure 3.

Step 3: In the OLPP method, the weight matrix W is traditionally computed using the heat kernel [31] to evaluate the similarity between connected vertices. However, in this paper, the heat kernel is replaced with cosine distance to better capture the angular relationship between feature vectors. This modification allows the edge weights to reflect the angular separation between features,which is more suitable for the characteristics of face

image data. For any pair of connected vertices i and j, the weight is calculated using cosine similarity, as formally defined in Equation (1):

$$\text{Cosine distance } (W) = \frac{x_j \cdot x_i}{\|x_j\| \, \|x_i\|} \tag{1}$$

Step 4: To obtain the orthogonal basis functions, a diagonal matrix D is first constructed, where each diagonal element D_{ii} represents the sum of the weights in the $i - th$ column of the weight matrix W, as shown in Equation (2):

$$D_{ii} = \sum_{j} W_{ji} \tag{2}$$

Subsequently, the graph Laplacian matrix L is computed using Equation (3):

$$L = D - W \tag{3}$$

Here, D is the diagonal degree matrix and W is the weight matrix constructed from cosine distances. The Laplacian matrix L is a symmetric and positive semi-definite matrix; its symmetry is inherent in its construction and does not require multiplication by its transpose [9]. The Laplacian matrix L acts as an operator on functions defined across the vertices of graph G [36]. When the

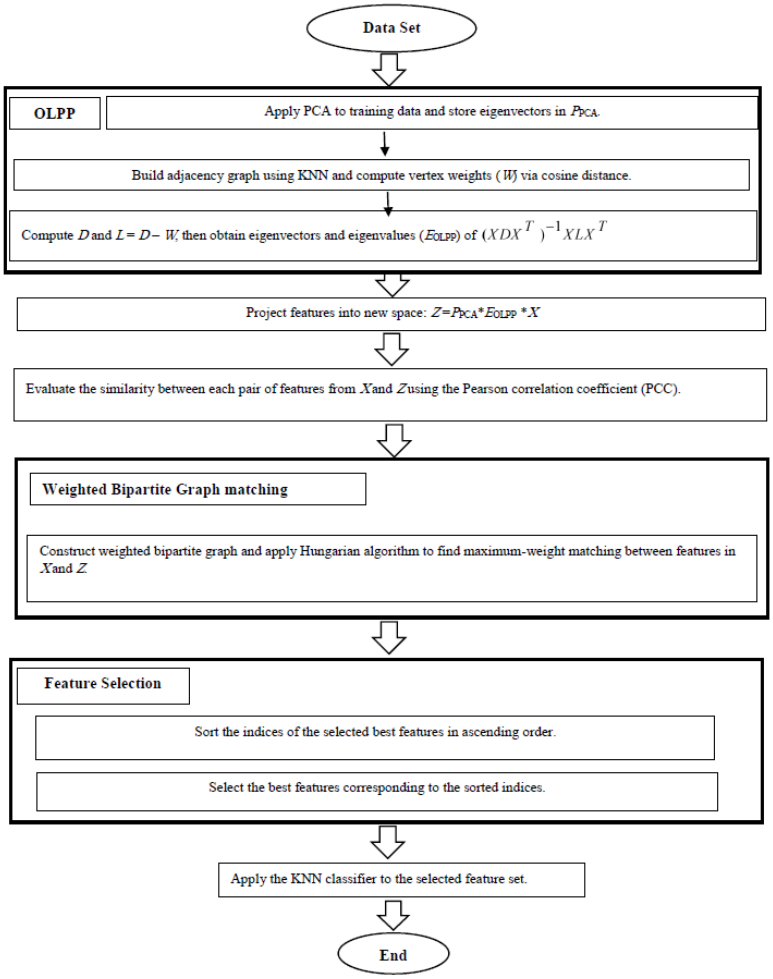


FIGURE 2. General flowchart of the proposed UFSOLPP method.

mapping is denoted as $f : M \to R$, the gradient $\nabla f(x) = \sum_{i=1}^{n} \frac{\partial f}{\partial x_i} \partial x_i$ can be interpreted as a vector field over the manifold. For sufficiently small values of ∂x, the following approximation holds, as shown in Equation (4):

$$|f(x + \delta x) - f(x)| \approx |\langle \nabla f(x), \delta x \rangle| \leq \|\nabla f\| \, \|\delta x\| \tag{4}$$

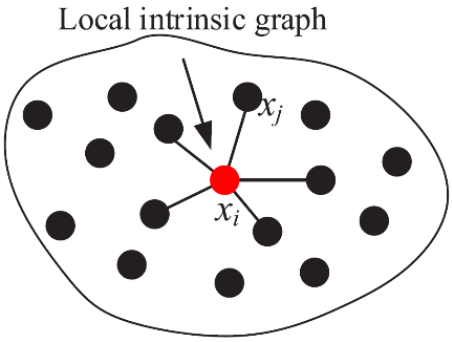


FIGURE 3. Local intrinsic graph [34].

If $\|\nabla f\|$ is minimized, then points near x are mapped to points close to f(x). Consequently, an optimal locality-preserving mapping can be obtained by minimizing the following objective function, as formulated in Equation (5):

$$\arg \min_{\|f\|_{L^2(M)}=1} \int_M \|\nabla f(x)\|^2 \, dx \tag{5}$$

This minimization $\int_M \|\nabla f(x)\|^2 \, dx$ directly corresponds to the objective function defined on the graph structure, as given in Equation (6):

$$Lf = \frac{1}{2} \sum_{i,j} (f_i - f_j)^2 w_{ij} \tag{6}$$

Minimizing the squared gradient corresponds to solving for the eigenfunctions of the Beltrami–Laplace operator L. The intrinsic relationship between the Beltrami–Laplace operator and the gradient is mathematically represented in Equation (7):

$$L = \operatorname{div} \nabla(f) \tag{7}$$

In this context, div denotes the divergence operator, which, when applied to the gradient field, validates the formulation presented in Equation (8):

$$\int_M \|\nabla f\|^2 = \int_M L(f) f \tag{8}$$

Here, f denotes an eigenfunction of the Laplacian operator L, which is utilized in the minimization process to preserve the local data structure $\int_M \|\nabla f(x)\|^2 \, dx$ [31].

Step 5: In the final step, the algorithm computes the eigenvectors and eigenvalues of the objective function defined in Equation (9) using the eigendecomposition technique.

$$(XDX^T)^{-1} XLX^T \tag{9}$$

Both XDX^T and XLX^T are symmetric and positive semi-definite matrices. In addition, the diagonal matrix D and the Laplacian matrix L are also symmetric and positive semi-definite [37]. A key characteristic of a positive semi-definite matrix is that all of its eigenvalues are non-negative, and eigenvectors corresponding to distinct eigenvalues are mutually orthogonal [9]. The eigenvectors are assembled into an orthogonal matrix E_1 , where each column corresponds to an eigenvector obtained from the eigendecomposition in Equation (10):

$$E_1 = [\, e_1; \ldots; e_n \,] \tag{10}$$

To project the original features into the new space, the matrix of eigenvectors from $\mathrm{P_{PCA}}$ is multiplied by matrix E_1, and the result is stored in U_1, as shown in Equation (11):

$$U_1 = P_{\mathrm{PCA}}\, E_1 \tag{11}$$

Finally, the output matrix Z is computed using Equation (12):

$$Z = U_1^T X \tag{12}$$

Since the OLPP algorithm in this study uses cosine distance, the resulting matrix Z may exhibit nonlinearity and may also contain complex values. To handle this issue, the complex conjugate of Z is first calculated. Subsequently, a linearized representation of the output is derived using Equation (13).

$$Z = \sqrt{\overline{Z}_1 Z_1} \tag{13}$$

It is important to emphasize that the algorithm utilizes all extracted features without performing any dimensionality reduction. Algorithm 1 summarizes the complete procedure of the OLPP algorithm.

3.2. **Calculating Similarity using the Pearson Correlation Coefficient (PCC).** In the second step, to evaluate the similarity between the original feature set X and the transformed set Z, the Pearson Correlation Coefficient is used as a similarity measure. Specifically, the correlation is computed between each feature in X and every feature in Z, and the resulting similarity values are encoded as weighted edges in a bipartite graph. The Pearson Correlation Coefficient (PCC), denoted by r, quantifies the strength of the linear relationship between two variables. Its value ranges from -1 to $+1$. $r \approx +1$ indicates a strong positive linear relationship, whereas $r \approx -1$ reflects a strong negative linear relationship [38]. Equation (14) calculates r:

$$r = \frac{\sum (Z - \overline{Z})(X - \overline{X})}{\sqrt{\sum (Z - \overline{Z})^2 (X - \overline{X})^2}} \tag{14}$$

If $\overline{Z}$ and $\overline{X}$ denote the means of sets Z and X, respectively, then when $r = 0$, Z and X are uncorrelated. The correlation distance between two sets Z and X is calculated using Equation (15):

$$CorrelationDistance(CD) = 1 - r \tag{15}$$

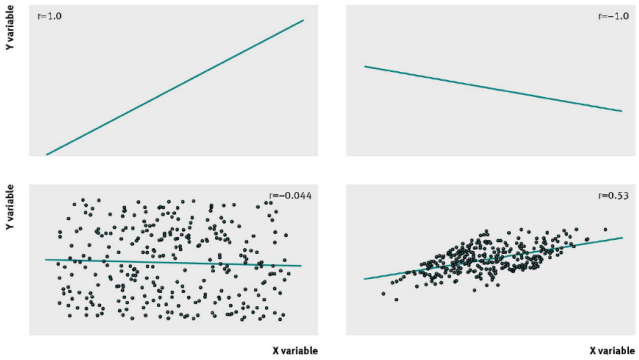


FIGURE 4. Scatter plots for different values of the Pearson correlation coefficient (r) [38].

To convert this distance into a similarity score, the correlation similarity (CS) is calculated by Equation (16) [9]:

$$CorrelationSimilarity(CS) = \frac{1}{2}(2 - CD) \tag{16}$$

The Pearson Correlation Distance (CD) takes values in the range 0 to 2, where CD = 0 indicates maximum similarity between the sets X and Z [9]. The resulting CS matrix defines the edge weights in the bipartite graph, which serves as the input to the Hungarian algorithm used for maximum matching. The PCC is scale-invariant because it is computed based on the mean and variance, making it robust to changes in measurement units. Given the presence of variance in both X and Z, PCC is an appropriate metric for comparing the original and transformed feature sets. Figure 4 demonstrates how effectively the PCC captures linear relationships between the orthogonal features in Z and the original features in X.

3.3. **Constructing the Weighted Bipartite Graph Matching.** In the third step of the proposed UFSOLPP algorithm, a weighted bipartite graph is constructed by considering the original features $X = \{x_1, x_2, x_3, \ldots, x_n\}$ and the transformed features $Z = \{z_1, z_2, z_3, \ldots, z_n\}$ as two disjoint sets, each containing n features. These sets form the two partitions of the graph, with edges between them weighted according to a similarity measure. Similarity evaluation between features is a fundamental task in many domains, particularly in image processing. Graph-based representations provide an effective framework for modeling objects and capturing complex relationships among them [39].

A graph is typically represented as $G = (V, E)$, where V denotes the set of vertices and E denotes the set of edges connecting pairs of vertices. A graph $G = (V, E)$ is considered a bipartite graph if its vertex set V can be divided into two disjoint subsets, A and B, such that no edge exists between vertices within

Algorithm 1 OLPP Algorithm

Input: Training data (X), $X \in \mathbb{R}^{m \times n}$.
Output: n optimal orthogonal features preserving the locality structure.
[1] **Begin Step 1: Apply PCA** Apply PCA to the training data X Store all eigenvectors in matrix $\P_{\mathrm{PCA}} \in \mathbb{R}^{n \times n}$ **Step 2: Construct an adjacency graph** Construct an adjacency graph using the original features Compute weight matrix $W \in \mathbb{R}^{m \times m}$ for graph edges using Equation (1) **Step 3: Compute Laplacian matrix** Compute diagonal matrix $D \in \mathbb{R}^{m \times m}$ using Equation (2) Compute the Laplacian matrix $L \in \mathbb{R}^{m \times m}$ as $L = D - W$ **Step 4: Solve eigenvalue problem** Solve the eigenvalue problem in Equation (9) Store eigenvectors in matrix E_1 **Step 5:** Compute projection matrix COMPUTE projection matrix U_1 as the product of $P_{\mathrm{PCA}}\,(n \times n)$ and $E_1\,(n \times m)$ **Step 6: Project original data** Project original data into a new feature space using $Z = U_1 X'\ Z$ **END**

the same subset [9]. The problem of graph matching, which aims to find an optimal correspondence between the vertices of two graphs, has received significant attention [39]. This problem involves selecting a subset of edges such that no two edges share a common vertex, with the objective of either maximizing similarity or minimizing discrepancies.

As previously mentioned, the two sets of original features and orthogonal features form the two partitions of a weighted bipartite graph. The weight of each edge is defined by the Pearson correlation coefficient between the connected features. Consequently, the weighted bipartite graph is defined as $G = (V, E)$, where $V = X \cup Z$ and E is the set of edges between them.

A matching is a subset $M \subseteq E$, such that for any two edges $e, e' \in M$, it holds that $e \neq e'$ and $e \cap e' = \emptyset$. A vertex not incident to any edge in M is called unmatched, and a perfect matching occurs when every vertex is matched to exactly one other vertex [9]. Two important definitions related to matching are as follows:

Definition 1: A maximum matching M in a graph is a matching with the maximum weight among all possible matchings [40].

Definition 2: An alternating path is a path in which the edges alternate between being part of the matching M and not being in M. An augmenting path is an alternating path that can increase the size of the current matching. It starts and ends at unmatched vertices, with intermediate vertices possibly matched or unmatched. The existence of an augmenting path indicates that the current matching is not maximal, and by swapping edges along this path, a larger matching can be obtained. When no augmenting path exists, the current matching is considered maximum (Berge, 1957) [40, 41].

In this work, the objective is to find a maximum weight matching in the constructed bipartite graph, where edge weights are determined by Pearson correlation coefficients between original and transformed features. Given the high

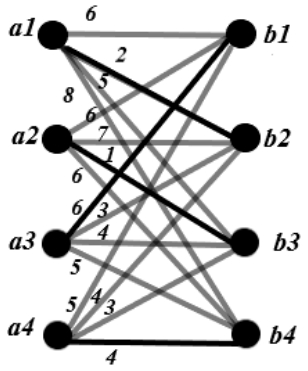


FIGURE 5. The weighted maximum matching on the bipartite graph is indicated by the thick edges [44].

dimensionality of the feature space, the problem is formulated in matrix form to maximize the total sum of the selected edge weights. To solve this efficiently, the Hungarian algorithm (also known as the Kuhn-Munkres algorithm) is employed [9,42]. The Hungarian algorithm is a well-known method for solving the maximum weight bipartite matching problem. It iteratively searches for augmenting paths and updates the matching M until no more augmenting paths exist. In this study, the augmenting path algorithm is used to identify a vertex cover Q that is equal in size to the bipartite graph $G_(u, v)$ and corresponds to the matching M. The total weight of the matching M, defined as the sum of the weights of its edges, is computed using Equation (17) [43]:

$$w(M) = \sum_{e \in M} w(e) \tag{17}$$

Figure 5 illustrates an example of a maximum-weight matching in a bipartite graph. Once the weighted bipartite graph is constructed, the Hungarian algorithm is applied to obtain an optimal perfect matching. The Hungarian algorithm improves the matching process by utilizing the augmenting path method. To enhance the current matching M on graph G, an augmenting path is initially identified. The matching is then expanded by replacing the edges of M with the edges of the augmenting path, which belong to $E-M$. This results in a new matching, M', that is larger than M [9]. The algorithm begins with an initial matching $M = \emptyset$. At each iteration, a directed graph $D = (X \cup Z, E')$ is constructed, where E' includes edges in M, directed from X to Z, and edges from $E-M$, directed from Z to X. If P is the augmenting path in the current matching M, then $M' = M\Delta P = (M - P) \cup (P - M)$. Equation (18) defines

the corresponding weight update or matching condition:

$$\begin{aligned} W(M') &= W(M) - W(P \cap M) + W(P - M) = W(M) - l(P) \\ l(P) &= W(P \cap M) - W(P - M) \\ |M'| &= |M| + 1 \end{aligned} \tag{18}$$

Let P represent the M-augmenting path, W denotes the total weight, and $l(P)$ indicates the length of the path P. The dual problem of matching is the vertex cover problem. In a graph $G = (V, E)$, a vertex cover is a subset $C \subseteq V$ such that every edge in G has at least one endpoint in C. This ensures that all edges in G are covered by the vertices in C. According to König's matching theorem, for any bipartite graph $G = (V, E)$, the size of a maximum matching equals the size of a minimum vertex cover [9]. Two key concepts in the Hungarian algorithm are the equality graph and labeling:

1. Equality Graph: An equality graph is derived from the cost matrix (the CS matrix from step 2). It includes edges whose weights become zero after row and column reductions. This graph forms the foundation for identifying a perfect matching.

2. Labeling: Labeling or vertex cover involves assigning values to rows and columns of the cost matrix in such a way that the resulting equality graph contains zero-weight edges. These labels are iteratively updated to enable and improve the construction of an optimal matching.

Let function $l : V \to \mathbb{R}$ denote a vertex labeling. A labeling is considered feasible if it satisfies the condition described in Equation (19). Based on this labeling, the equality graph $G = (V, E_l)$ is defined as shown in Equation (20) [45].

$$l(z) + l(x) \geq w(z, x), \quad \forall x \in X,\ z \in Z \tag{19}$$

$$E_l = \{(y, x) : l(y) + l(x) = w(x, y)\} \tag{20}$$

Equation (20) states that an edge (y, x) belongs to the equality graph $G_(u, v)$ if and only if the sum of the labels $l(y)$ and $l(x)$ of vertices y and x, respectively, is equal to the weight $w(x, y)$ of the edge connecting them. The equality graph is then used to identify a perfect matching by selecting edges that satisfy this condition. According to the Kuhn–Munkres theorem, if M is a perfect matching in the equality graph E_l and l is a feasible labeling, then M is a maximum weighted matching [45]. Algorithm 2 outlines the step-by-step procedure of the Hungarian algorithm for solving the maximum-weight matching problem using a matrix-based representation. Finally, the Hungarian algorithm is applied to the CS matrix obtained in Step 2. By iteratively updating vertex labels and constructing equality graphs, the Hungarian algorithm identifies an optimal set of matched features. The selected features, corresponding to the vertices in X, are then used for classification using a k-nearest neighbors (KNN) classifier.

3.4. **The proposed UFSOLPP algorithm.** Algorithm 3 outlines the step-by-step process of the proposed UFSOLPP algorithm. To aid comprehension, Figure 6 illustrates each step of the UFSOLPP algorithm using a sample dataset. The bolded numbers in the figure indicate the features selected by the method.

4. RESULTS

This section presents a comparative evaluation of the proposed UFSOLPP method in comparison with eight well-known unsupervised feature selection algorithms, namely UFSPCA-2022 [9], AEFS-2018 [20], DGUFS-2018 [21], SCEFS-2021 [22], SCAFS-2021 [22], RSR-2015 [23], SRCFS-2019 [24], and URAFS-2018 [25]. All algorithms were tested on five widely used image datasets: Jaffe, Yale, ORL, COIL-20, and pixraw10P. Experiments were conducted using MATLAB R2018b on a system equipped with an Intel Core i7 processor, 8 GB of RAM, and Microsoft Windows 10.

4.1. **Datasets description.** To evaluate the performance of the proposed UFSOLPP method against the eight aforementioned algorithms, experiments were conducted on five benchmark image datasets. A summary of these datasets is provided in Table 1. The datasets are described as follows: JAFFE [46]: Contains facial expression images of Japanese female subjects. Yale [46]: Consists of 165 grayscale facial images of 15 individuals in GIF format. ORL [47]: Includes facial images of 40 individuals with variations in expressions and facial details. Pixraw10P [47]: Comprises face images captured under varying lighting conditions. COIL-20 dataset [48]: Contains images of 20 different objects, each captured from multiple viewpoints.

4.2. **Classifier.** Classification plays a crucial role in big data analysis, data science, and machine learning. In this study, the K-Nearest Neighbors (KNN) classifier was used to evaluate the performance of the selected features on the test data samples. KNN is widely recognized for its simplicity and effectiveness. As a nonparametric algorithm, it classifies test samples by measuring their distances to training samples, making it particularly suitable for classification tasks in big data scenarios [49]. The core principle of KNN is that a test sample is likely to belong to the same class as its nearest neighbors in the feature space [50]. To predict the class label of a new sample, KNN identifies the k nearest neighbors in the training dataset using a specified distance metric and assigns the most frequent class label among these neighbors to the new sample [9]. In this study, the value of k was set to 5.

Algorithm 2 Algorithm 2: The Hungarian Algorithm to find the maximum weight bipartite graph matching

Input: n-by-n correlation similarity (CS) matrix, n is the number of features. **Output:** Optimal solution: n index of features that have maximum total weight matching. [1] **BEGIN**
// Step 1: Initialize labels FOR each row i in set X DO u[i] $\leftarrow$ max(CS[i][j]) for all j END FOR FOR each column j in set Z DO $v[j] \leftarrow 0$ END FOR
// Step 2: Construct initial equality subgraph $G(u, v)$ WHILE no perfect matching M exists in the equality subgraph DO
// Step 3: Cover zeros in CS using a minimum number of lines Construct matching M on zero entries of CS IF M is perfect THEN BREAK END IF
// Step 4: Identify uncovered rows R and uncovered columns T Determine sets R and T (uncovered rows and columns)
// Step 5: Adjust matrix using the smallest uncovered value ε $\varepsilon \leftarrow$ minimum uncovered value in CS
FOR each $i \in R$ DO FOR $j = 1$ to n DO $CS[i][j] \leftarrow CS[i][j] - \varepsilon$ END FOR END FOR
FOR each $j \in T$ DO FOR $i = 1$ to n DO $CS[i][j] \leftarrow CS[i][j] + \varepsilon$ END FOR END FOR
END WHILE
RETURN M **END**

Algorithm 3 Proposed UFSOLPP Algorithm

Input: Training dataset $X' \in \mathbb{R}^{n \times m}$.
Output: Indices of selected features and total maximum weight
Begin Calculate new features $Z \in \mathbb{R}^{n \times m}$ **using Algorithm 1**
For$i = 1$ **to** n **For**$j = 1$ **to** n **C**ompute Pearson Correlation Distance $CD \in \mathbb{R}^{n \times n}$ using `pdist2(X', Z, 'correlation')` **C**alculate Correlation Similarity $CS \in \mathbb{R}^{n \times n}$ as:

$$CS = \frac{2 - CD}{2}$$

End For End For Construct a Weighted Bipartite Graph $G = (V, E)$ based on the CS matrix **A**pply the Hungarian Algorithm on the CS matrix to obtain selected feature indices C and total maximum weight T using Algorithm 2 **Return:** Selected feature indices $C \in \mathbb{R}^{1 \times n}$ and total maximum weight T **END**

Table 1. Details of the datasets.

Dataset	# Samples	# Features	# Classes	Domain
Jaffe	213	676	7	Image, face
Yale	165	1024	15	Image, face
ORL	400	1024	40	Image, face
COIL-20	1440	1024	20	Image, object
Pixraw10P	100	10000	10	Image, face

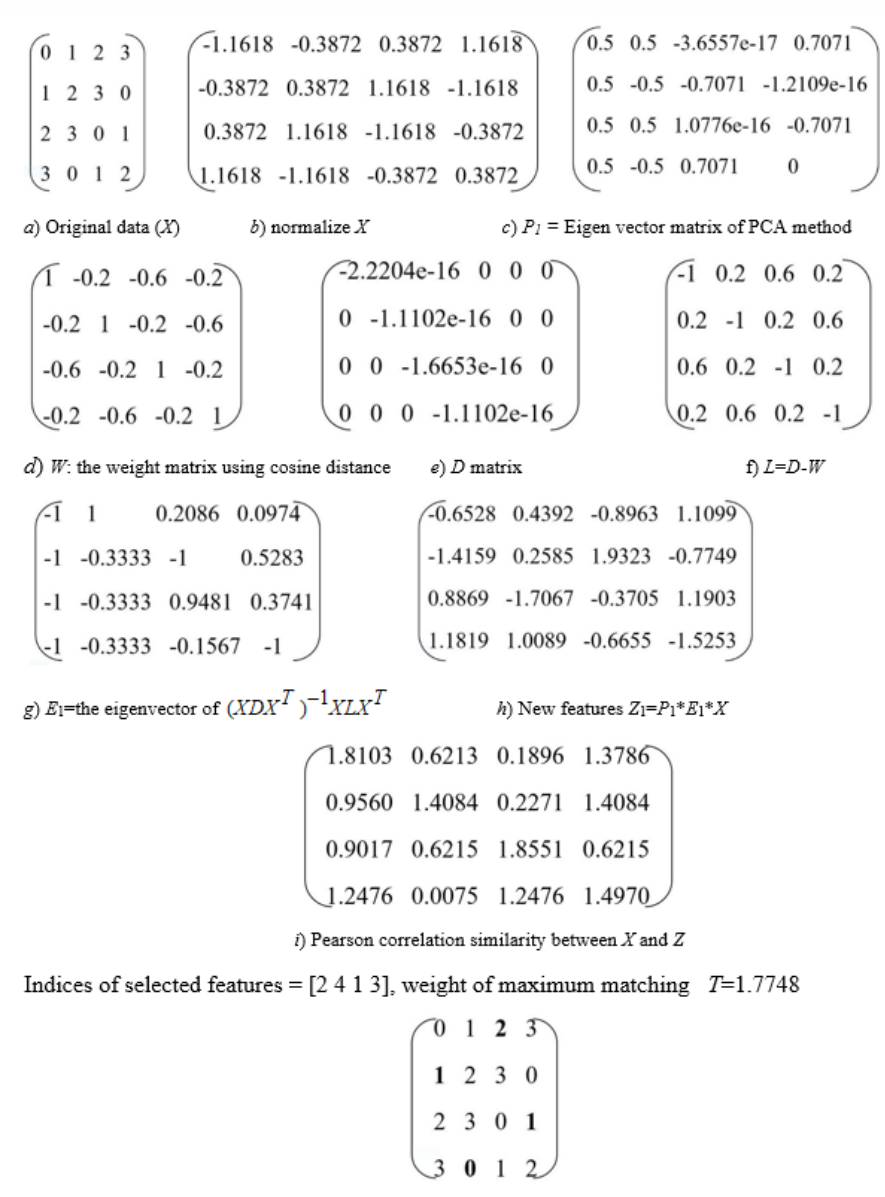


FIGURE 6. Step-by-step solution of an example using the proposed UFSOLPP algorithm.

4.3. **Performance Evaluation Criteria.** In this study, five performance metrics were used to evaluate the effectiveness of the proposed UFSOLPP

method compared to the eight aforementioned methods. These metrics include Accuracy, Recall, Precision, Normalized Mutual Information (NMI), and F-measure. A brief overview of each criterion is provided below. In all metrics, True Positives (TP) and True Negatives (TN) refer to the correctly classified positive and negative samples, respectively. Conversely, False Positives (FP) and False Negatives (FN) refer to misclassified positive and negative samples [9].

Accuracy: The proportion of correctly classified instances out of all instances. It is defined as Equation (21) [51]:

$$\text{Accuracy} = \frac{TN + TP}{TN + FN + TP + FP} \tag{21}$$

Precision: The proportion of correctly predicted positive samples among all samples predicted as positive (Equation 22) [9]:

$$Precision = \frac{TP}{TP + FP} \tag{22}$$

Recall, also known as Sensitivity or True Positive Rate: The proportion of correctly predicted positive samples out of all actual positive samples (Equation 23) [9]:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{23}$$

F-measure (or F1-score): A harmonic mean of Precision and Recall, providing a balanced measure (Equation 24) [9]:

$$F\text{-measure} = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{24}$$

Normalized Mutual Information (NMI) [9]: Measures the similarity between predicted and actual class labels, accounting for entropy in both sets. It is computed using Equations (25-27). Where MI(Y, X) is mutual information between labels, and $H(Y)$ and $H(X)$ denote their respective entropies:

$$H(Y) = -E[\log p(y)] = -\sum\nolimits_{y} p(y) \log p(y) \tag{25}$$

Here, p (y) represents the probability of occurrence of the i-th class. NMI quantifies the mutual information between the predicted and actual class labels while considering the uncertainty associated with each label set. The (MI) for Y and X is calculated by Equation (26) [9].

$$I(X;Y) = H(X) - H(X \mid Y) \tag{26}$$

The entropies of Y and X are denoted by $H(Y)$ and $H(X)$, respectively. NMI is calculated using Equation (27).

$$H(Y) = -E[\log p(y)] = -\sum\nolimits_{y} p(y) \log p(y) \tag{27}$$

NMI values range between 0 and 1. A value of 0 indicates no mutual information (complete disagreement), while a value of 1 indicates perfect agreement [9].

In all evaluations, the performance of the proposed UFSOLPP method is compared with the baseline methods using 30 independent runs for each algorithm on each dataset. For each run, the dataset is randomly split: 70% of the samples for training and 30% for testing, ensuring statistical robustness.

In each run, algorithms select the top-m most representative features, where m ranges from 10 to 100 in increments of 10. The k-nearest neighbor (KNN) classifier is then applied to these selected features.

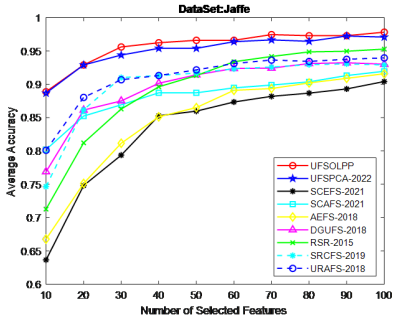
Figures 6–10, the horizontal axis represents the number of selected features, ranging from 10 to 100 in steps of 10, showing how classification performance changes with different feature subset sizes.

To compare methods statistically, the Wilcoxon signed-rank test [5] is applied. This nonparametric test evaluates pairwise differences in performance across all runs. For each pair of methods, the comparison result is denoted as follows:

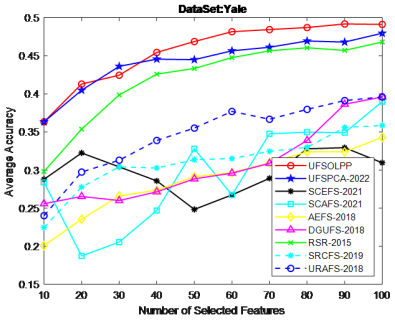
- "+" : UFSOLPP performs significantly better
- "−" : UFSOLPP performs significantly worse
- "≈" : No significant difference

Figure 7 illustrates the accuracy results of UFSOLPP versus state of the art methods across the five datasets. As shown, UFSOLPP consistently delivers superior performance, especially on the Yale and ORL datasets, which contain complex nonlinear facial structures. Its ability to preserve intrinsic geometry significantly contributes to its high classification performance. Notably, on the high dimensional Pixraw10P dataset (10,000 features), UFSOLPP achieves over 80% accuracy, demonstrating its robustness and efficiency in handling noisy, high dimensional data.
Overall, the results confirm that UFSOLPP is highly effective in capturing meaningful features, maintaining stability across datasets, and enhancing classification performance, especially in nonlinear and high-dimensional scenarios.

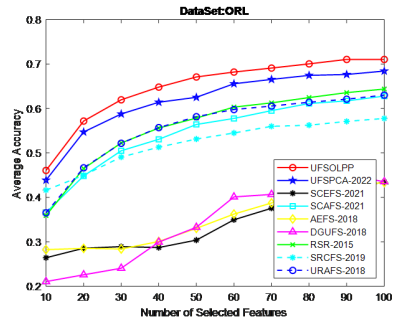
Figure 8 presents the average precision performance of the proposed UFSOLPP algorithm compared to other unsupervised feature selection methods. The results clearly demonstrate that UFSOLPP consistently achieves the highest precision on ORL, Yale, and Jaffe datasets, all of which exhibit nonlinear manifold structures resulting from variations in facial expressions and poses. This outcome highlights the algorithm's effectiveness in identifying the most relevant and discriminative features in such complex scenarios. Notably, UFSOLPP also performs exceptionally well on the high-dimensional Pixraw10P dataset. Despite its complexity, the proposed UFSOLPP method consistently ranks among the top-performing methods, achieving first or second place across



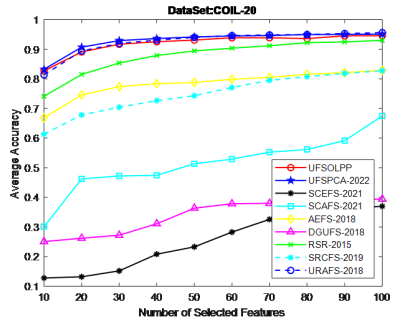
(A) Jaffe



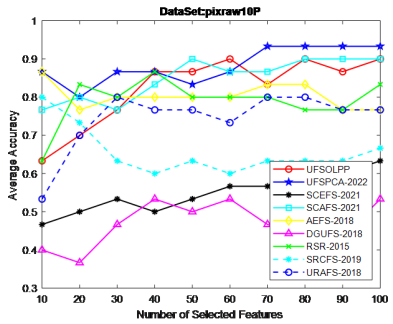
(B) Yale



(C) ORL

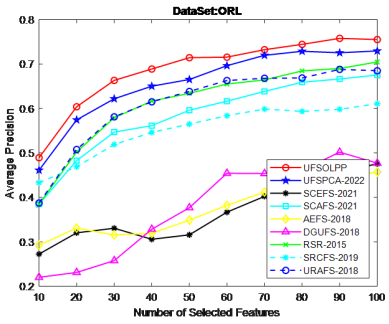


(D) COIL-20

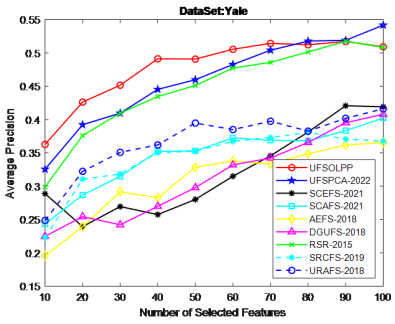


(E) pixraw10P

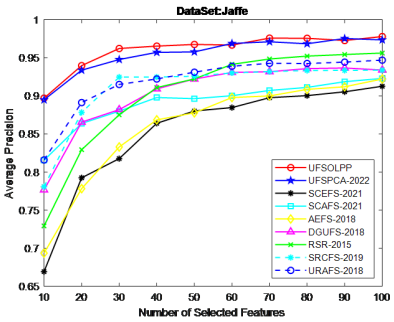
FIGURE 7. Average Accuracy Comparison of UFSOLPP with Other Methods across Five Datasets



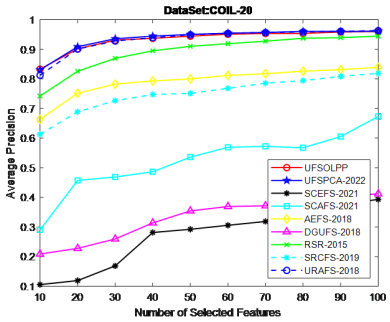
(A) Jaffe



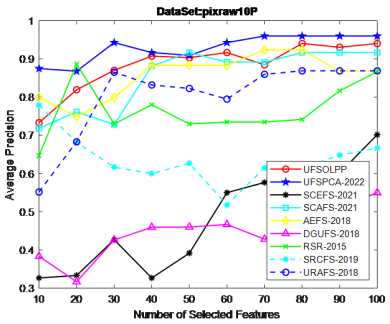
(B) Yale



(C) ORL



(D) COIL-20



(E) pixraw10P

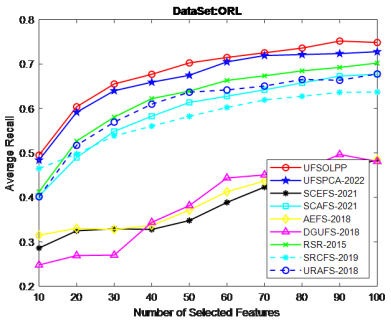
FIGURE 8. Average Precision Comparison of UFSOLPP with Other Methods across Five Datasets

multiple feature subset sizes. These results demonstrate UFSOLPP's robustness in high-dimensional spaces and its ability to retain precision while avoiding the selection of irrelevant or noisy features. Overall, the findings confirm the adaptability and effectiveness of UFSOLPP in processing both facial and non-facial image datasets with diverse structural properties, reinforcing its superiority over existing methods.

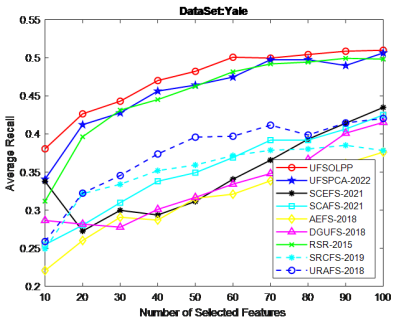
Figure 9 depicts the average recall performance of the proposed UFSOLPP algorithm in comparison with other unsupervised feature selection methods across five datasets. As shown, UFSOLPP consistently achieves strong recall, particularly on facial image datasets such as Jaffe and ORL. On the Jaffe dataset, it attains the highest recall of 0.9604, outperforming UFSPCA (0.9544) and URAFS (0.9158), highlighting its superior capability in capturing a broader set of relevant features amid facial expression variations. Similarly, for the ORL dataset, UFSOLPP achieves a recall of 0.6811, exceeding UFSPCA (0.6647) and URAFS (0.6035), which underscores its effectiveness in handling datasets with more classes and complex facial structures. While on the COIL20 and Pixraw10P datasets, UFSPCA slightly outperforms UFSOLPP, still maintains competitive recall scores (0.9230 and 0.8865, respectively), demonstrating robustness even in object image data and high-dimensional feature spaces. These results confirm UFSOLPP's strength in selecting a broader set of relevant features from nonlinear manifold structures across diverse datasets.

Figure 10 illustrates the average F-measure performance of the proposed UFSOLPP algorithm compared to other unsupervised feature selection methods across five benchmark datasets. The F-measure, which balances precision and recall, is a key indicator of the algorithm's overall effectiveness in selecting relevant features while minimizing both false positives and false negatives. UFSOLPP achieves the highest F-measure on facial image datasets such as Jaffe (0.9602) and ORL (0.6832), reflecting its effectiveness in handling nonlinear relationships and subtle variations in facial expressions and structures. On the Yale dataset, UFSOLPP also performs best (0.4729), demonstrating its robustness in handling variations in lighting, pose, and expression. Moreover, UFSOLPP maintains strong F-measure values on non-facial datasets like COIL20 (0.9277) and Pixraw10P (0.8855), indicating its adaptability to high-dimensional object image data. These results highlight UFSOLPP's overall versatility and effectiveness across both facial and general image datasets.

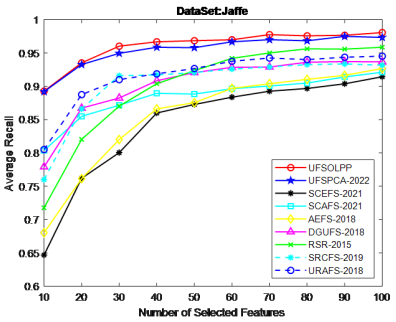
Figure 11 illustrates the average Normalized Mutual Information (NMI) obtained by the proposed UFSOLPP algorithm across five datasets. The results highlight UFSOLPP's strong capability in selecting relevant features for unsupervised learning, particularly on facial image datasets such as Yale and ORL. Additionally, its competitive performance on the COIL20 object image dataset demonstrates the algorithm's adaptability to non-facial domains. These findings confirm UFSOLPP's robustness in handling datasets with nonlinear manifolds and high dimensionality.



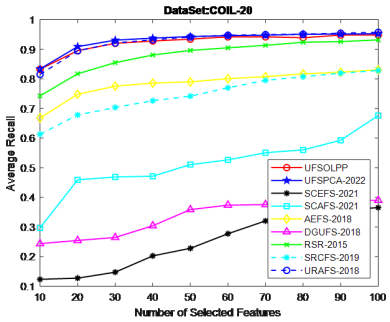
(A) Jaffe



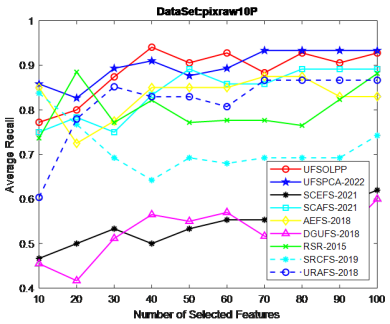
(B) Yale



(C) ORL

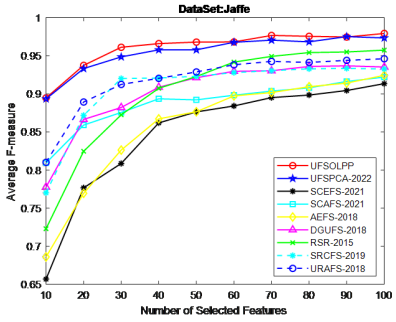


(D) COIL-20

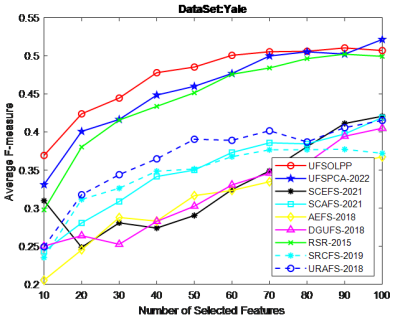


(E) pixraw10P

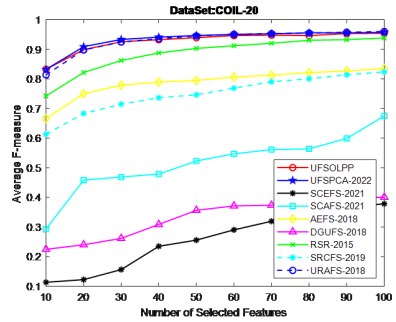
FIGURE 9. Average Recall Comparison of UFSOLPP with Other Methods across Five Datasets



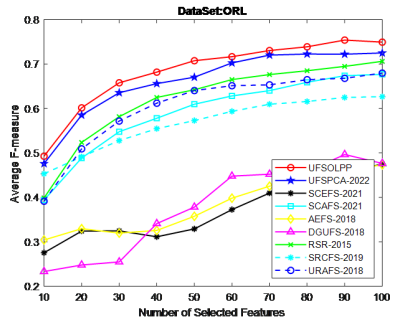
(A) Jaffe



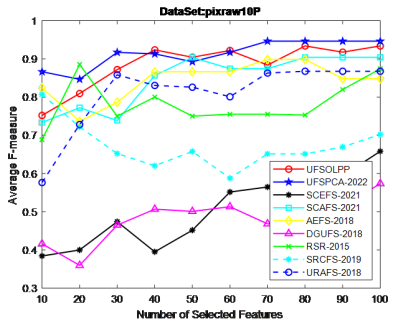
(B) Yale



(C) ORL

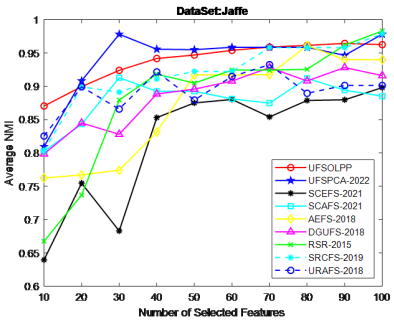


(D) COIL-20

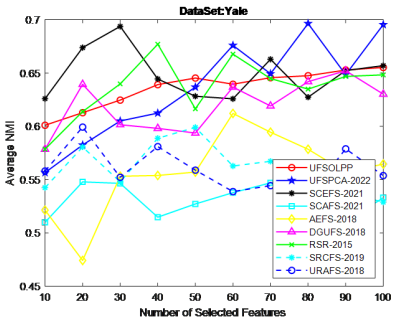


(E) pixraw10P

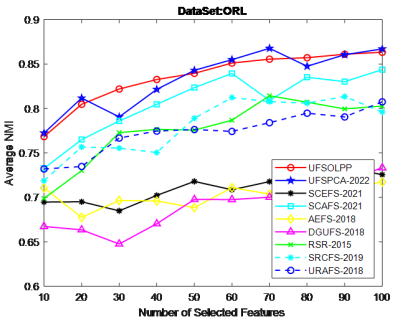
FIGURE 10. Average F-measure Comparison of UFSOLPP with Other Methods across Five Datasets



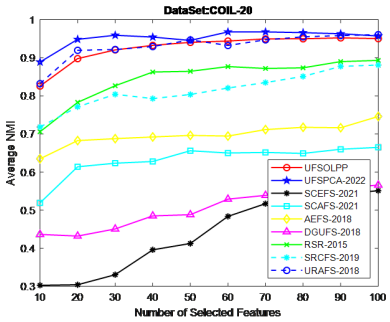
(A) Jaffe



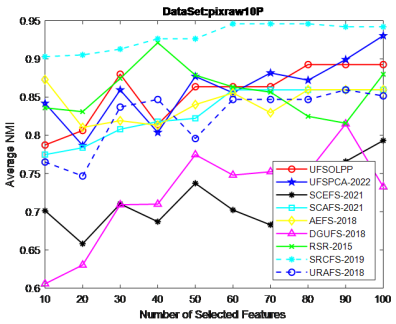
(B) Yale



(C) ORL



(D) COIL-20



(E) pixraw10P

FIGURE 11. Average NMI Comparison of UFSOLPP with Other Methods across Five Datasets

Table 2 summarizes the average classification accuracy of the proposed UFSOLPP algorithm in comparison with several state of the art unsupervised feature selection methods across five benchmark datasets: Jaffe, Yale, ORL, COIL20, and Pixraw10P. These datasets, particularly those comprising facial images, exhibit nonlinear manifold structures due to variations in facial expressions, poses, and illumination. These variations make the feature selection process significantly more challenging.

• **Jaffe Dataset:** This dataset includes 213 samples with 676 features across 7 expression-based classes. UFSOLPP achieved the highest accuracy of 0.9568, outperforming UFSPCA (0.9506) and URAFS (0.9103). This improvement of 0.62% over the next best method highlights UFSOLPP's superior capacity to extract highly discriminative features from nonlinear facial manifolds.

• **Yale Dataset:** With 165 samples, 1024 features, and 15 classes, this dataset is characterized by more complex facial variations. UFSOLPP attained an accuracy of 0.4560, exceeding UFSPCA (0.4308) and RSR (0.4295) by 2.52% and 2.65%, respectively. These results suggest the proposed method's robustness in handling datasets with intricate feature distributions and relatively low interclass separability.

• **ORL Dataset:** This dataset poses a significant challenge due to its 40 class setup. UFSOLPP outperformed all other methods with an accuracy of 0.6467, compared to 0.6171 for UFSPCA and 0.5562 for URAFS. The notable margin (2.96% over UFSPCA and 9.05% over URAFS) further demonstrates the algorithm's effectiveness in selecting discriminative features from complex, high-dimensional image data.

• **COIL20 Dataset:** As an object dataset with 1440 samples and 1024 features, COIL20 differs structurally from facial datasets. While UFSPCA achieved slightly higher accuracy (0.9290) than UFSOLPP (0.9198), the proposed method still ranks among the top-performing algorithms and outperforms SCEFS (0.2537) and DGUFS (0.3385) by significant margins, demonstrating adaptability across domains.

• **Pixraw10P Dataset:** This dataset presents a severe high-dimensional challenge with 10,000 features and only 100 samples. UFSOLPP achieved an accuracy of 0.8233, slightly lower than UFSPCA (0.8833) but higher than other methods such as DGUFS (0.4667). Despite the minor drop in performance compared to UFSPCA, UFSOLPP remains competitive and proves capable of handling ultra-high-dimensional spaces effectively.

In summary, the proposed UFSOLPP algorithm consistently outperforms or competes closely with existing methods across all datasets. Its ability to model nonlinear structures allows it to achieve top results in datasets like Jaffe and ORL, where facial expressions generate highly curved manifolds. The Wilcoxon signed-rank test confirms that the observed improvements are statistically significant, further supporting the superiority of UFSOLPP in extracting informative and compact feature subsets from high-dimensional data. The best-performing results in Table 2 are bolded for clarity.

Table 2. Average Accuracy Comparison of UFSOLPP with Other Methods on Five Datasets

Dataset	UFSOLPP	UFSPCA	SCEFS	SCAFS	AEFS	DGUFS	RSR	SRCFS	URAFS
Jaffe	**0.9568**	0.9506	0.8331	0.8829	0.8459	0.8963	0.8924	0.8987	0.9103
Yale	**0.4560**	0.4308	0.3273	0.3253	0.2941	0.3170	0.4295	0.3271	0.3523
ORL	**0.6467**	0.6171	0.3401	0.5444	0.3501	0.3420	0.5602	0.5218	0.5562
COIL20	0.9198	**0.9290**	0.2537	0.5135	0.7831	0.3385	0.8778	0.7487	0.9254
Pixraw10P	0.8233	**0.8833**	0.5500	0.8500	0.8033	0.4667	0.7900	0.6567	0.7433
Wilcoxon	+	+	+	+	+	+	+		

Table 3. Average Precision Comparison of UFSOLPP with Other Methods on Five Datasets

Dataset	UFSOLPP	UFSPCA	SCEFS	SCAFS	AEFS	DGUFS	RSR	SRCFS	URAFS
Jaffe	**0.9600**	0.9548	0.8524	0.8913	0.8591	0.9024	0.9019	0.9097	0.9191
Yale	**0.4782**	0.4599	0.3218	0.3447	0.3086	0.3134	0.4462	0.3415	0.3663
ORL	**0.6866**	0.6574	0.3670	0.5829	0.3723	0.3763	0.6117	0.5518	0.6104
COIL20	0.9324	**0.9363**	0.2686	0.5227	0.7917	0.3298	0.8911	0.7503	0.9323
Pixraw10P	0.8847	**0.9295**	0.4827	0.8540	0.8581	0.4426	0.7668	0.6367	0.8017
Wilcoxon		+	+	+	+	+	+	+	+

Table 3 presents the average precision achieved by the proposed UFSOLPP algorithm compared to other unsupervised feature selection methods across five distinct datasets: Jaffe, Yale, ORL, COIL20, and Pixraw10P. Precision is a vital metric in evaluating the relevance of selected features, especially in datasets with nonlinear manifold structures, such as facial image datasets.

• **Jaffe Dataset:** UFSOLPP achieved the highest precision of 0.9600, surpassing UFSPCA (0.9548) and URAFS (0.9191). This result underscores UFSOLPP's superior ability to select relevant features from facial images, which include variations in expressions, and highlights its effectiveness in capturing nonlinear manifold structures inherent to facial data.

• **Yale Dataset:** In this more complex dataset, UFSOLPP achieved a precision of 0.4782, outperforming UFSPCA (0.4599) and RSR (0.4462). This demonstrates the algorithm's capability to handle larger feature spaces and complex facial variations, further validating its robustness in high-dimensional datasets.

• **ORL Dataset:** UFSOLPP achieved a precision of 0.6866, outperforming UFSPCA (0.6574) and URAFS (0.6104). This higher precision indicates that UFSOLPP is particularly effective in identifying key features in datasets with a larger number of classes and more complex facial structures.

• **COIL20 Dataset:** On this object image dataset, UFSOLPP attained a precision of 0.9324, which is slightly lower than UFSPCA (0.9363) but still superior to other methods like SCEFS (0.2686) and DGUFS (0.3298). This demonstrates the versatility of UFSOLPP, which excels beyond facial image data by effectively handling object datasets.

• **Pixraw10P Dataset:** With 10,000 features, this high-dimensional dataset

presented a significant challenge. UFSOLPP achieved a precision of 0.8847, slightly lower than UFSPCA (0.9295) but outperforming algorithms like SCAFS (0.8540) and AEFS (0.8581). This result further emphasizes UFSOLPP's strong feature selection capability, even in the context of very high-dimensional datasets.

In summary, the proposed UFSOLPP algorithm consistently outperforms or matches other methods in precision across various datasets, particularly excelling in facial image datasets such as Jaffe and ORL. These results highlight the algorithm's effectiveness in selecting relevant features from nonlinear manifolds, particularly in high-dimensional data. The Wilcoxon test results reinforce the significance of UFSOLPP's improvements in precision over other methods. The best results in Table 3 are highlighted for clarity.

TABLE 4. Average Recall Comparison of UFSOLPP with Other Methods on Five Datasets

Dataset	UFSOLPP	UFSPCA	SCEFS	SCAFS	AEFS	DGUFS	RSR	SRCFS	URAFS
Jaffe	**0.9604**	0.9544	0.8434	0.8845	0.8558	0.9025	0.8999	0.9032	0.9158
Yale	**0.4725**	0.4566	0.3463	0.3516	0.3121	0.3328	0.4512	0.3509	0.3738
ORL	**0.6811**	0.6647	0.3819	0.5916	0.3951	0.3846	0.6199	0.5767	0.6035
COIL20	0.9230	**0.9303**	0.2487	0.5115	0.7848	0.3330	0.8792	0.7484	0.9265
Pixraw10P	0.8965	**0.8992**	0.5433	0.8400	0.8310	0.5195	0.8010	0.7132	0.8169
Wilcoxon		+	+	+	+	+	+	+	+

Table 4 presents the average recall achieved by the proposed UFSOLPP algorithm, compared with other unsupervised feature selection methods across five datasets: Jaffe, Yale, ORL, COIL20, and Pixraw10P. Recall is a critical metric for assessing the algorithm's ability to select all relevant features, particularly in complex, high-dimensional datasets characterized by nonlinear manifold structures.

• **Jaffe Dataset:** UFSOLPP achieved the highest recall of 0.9604, outperforming UFSPCA (0.9544) and URAFS (0.9158). This superior recall emphasizes UFSOLPP's ability to capture a broader set of relevant features, especially in the context of the nonlinear facial expression variations in the Jaffe dataset.

• **Yale Dataset:** UFSOLPP attained a recall of 0.4725, surpassing UFSPCA (0.4566) and RSR (0.4512). This result indicates the algorithm's robustness in managing complex facial variations in the Yale dataset, which includes changes in lighting, expressions, and pose factors that typically challenge feature selection.

• **ORL Dataset:** UFSOLPP achieved a recall of 0.6811, outperforming UFSPCA (0.6647) and URAFS (0.6035). This high recall score demonstrates UFSOLPP's effectiveness in selecting a larger portion of relevant features, even in datasets that contain a greater number of classes and more intricate facial variations.

• **COIL20 Dataset**: For the COIL20 object image dataset, UFSOLPP achieved a recall of 0.9230, slightly lower than UFSPCA (0.9303), but still

outperforming other algorithms like SCEFS (0.2487) and DGUFS (0.3330). This result illustrates UFSOLPP's ability to effectively select relevant features in both facial and object image datasets.

• **Pixraw10P Dataset:** In this high-dimensional dataset with 10,000 features, UFSOLPP achieved a recall of 0.8865, slightly lower than UFSPCA (0.8992), but significantly outperforming other methods like SCAFS (0.84) and AEFS (0.8310). Despite the high-dimensional space, UFSOLPP shows its strength in efficiently selecting the majority of relevant features.

In conclusion, the proposed UFSOLPP algorithm demonstrates strong recall performance across multiple datasets, particularly in facial image datasets like Jaffe and ORL, where capturing nonlinear relationships is essential. The algorithm's ability to achieve high recall scores indicates that it can effectively select a comprehensive set of relevant features in both high-dimensional and complex datasets. The Wilcoxon test results further confirm that UFSOLPP consistently outperforms other algorithms in terms of recall, solidifying its effectiveness for feature selection in manifold-based data structures. The best results in Table 4 are highlighted for clarity.

Table 5. Average F-measure Comparison of UFSOLPP with Other Methods on Five Datasets

Dataset	UFSOLPP	UFSPCA	SCEFS	SCAFS	AEFS	DGUFS	RSR	SRCFS	URAFS
Jaffe	**0.9602**	0.9546	0.8476	0.8877	0.8572	0.9024	0.9008	0.9063	0.9173
Yale	**0.4729**	0.4562	0.3289	0.3483	0.3069	0.3186	0.4436	0.3442	0.3664
ORL	**0.6832**	0.6617	0.3706	0.5897	0.3837	0.3787	0.6200	0.5671	0.6043
COIL20	0.9277	**0.9333**	0.2568	0.5169	0.7881	0.3309	0.8851	0.7492	0.9294
Pixraw10P	0.8855	**0.9140**	0.5063	0.8467	0.8441	0.4774	0.7833	0.6723	0.8088
Wilcoxon		+	+	+	+	+	+	+	+

Table 5 presents the average F-measure values for the proposed UFSOLPP algorithm in comparison with several other unsupervised feature selection methods across five benchmark datasets. The F-measure, which harmonizes precision and recall, serves as a crucial metric for assessing the overall performance of feature selection algorithms, particularly in datasets characterized by nonlinear structures such as facial images.

• **Jaffe Dataset:** UFSOLPP achieved the highest F-measure (0.9602), outperforming UFSPCA (0.9546) and URAFS (0.9173). This result indicates that UFSOLPP is highly effective in balancing precision and recall by selecting relevant features while minimizing false positives and false negatives, especially in datasets with facial expression variations.

• **Yale Dataset:** On the Yale dataset, UFSOLPP attained an F-measure of 0.4729, surpassing UFSPCA (0.4562) and RSR (0.4436). Given the complexity of this dataset featuring variations in lighting, pose, and facial expression, UFSOLPP's performance reflects its robustness in identifying features that capture these subtle differences.

• **ORL Dataset:** With an F-measure of 0.6832, UFSOLPP outperformed

UFSPCA (0.6617) and URAFS (0.6043). This high score demonstrates the algorithm's effectiveness in managing the trade-off between precision and recall in datasets with a large number of facial classes and structural variations.
• **COIL20 Dataset:** UFSOLPP achieved an F-measure of 0.9277, which is slightly lower than UFSPCA (0.9333) but notably higher than methods such as SCEFS (0.2568) and DGUFS (0.3309). This outcome showcases UFSOLPP's adaptability and strong performance not only in facial image datasets but also in object recognition tasks.
• **Pixraw10P Dataset:** UFSOLPP obtained an F-measure of 0.8855, slightly below UFSPCA (0.9140), yet it significantly outperformed SCEFS (0.5063) and SCAFS (0.8467) and other methods compared. Despite the high dimensionality and complexity of this dataset, UFSOLPP maintained competitive performance, underlining its efficiency in selecting informative features.
Overall, the proposed UFSOLPP algorithm demonstrates consistently high F-measure values across diverse datasets, particularly excelling in facial image datasets like Jaffe and ORL, where capturing nonlinear patterns is crucial. By achieving strong performance across both facial and object image datasets, UFSOLPP suggests that a robust and versatile feature selection method. Furthermore, the Wilcoxon test results statistically validate its superiority in terms of F-measure, confirming its capability in selecting relevant features from high-dimensional and complex data distributions.

TABLE 6. Average NMI Comparison of UFSOLPP with Other Methods on Five Datasets

Dataset	UFSOLPP	UFSPCA	SCEFS	SCAFS	AEFS	DGUFS	RSR	SRCFS	URAFS
Jaffe	**0.9384**	0.9224	0.8197	0.8791	0.8728	0.8844	0.8827	0.9203	0.8932
Yale	0.6363	0.6359	**0.6491**	0.5300	0.5567	0.6191	0.6368	0.5617	0.5599
ORL	**0.8356**	0.8337	0.7102	0.8070	0.7031	0.6901	0.7765	0.7807	0.7736
COIL20	0.9264	**0.9518**	0.4362	0.6314	0.6979	0.5023	0.8448	0.8155	0.9303
Pixraw10P	0.8557	**0.8608**	0.7195	0.8305	0.8421	0.7230	0.8582	0.9296	0.8244
Wilcoxon		+	+	+	+	+	+	+	+

Table 6 presents the average Normalized Mutual Information (NMI) for the proposed UFSOLPP algorithm, comparing it's performance with several other unsupervised feature selection methods across five datasets. NMI quantifies the agreement between predicted and true class labels, making it a critical metric for evaluating feature selection effectiveness in unsupervised learning.
Jaffe Dataset: UFSOLPP achieved the highest NMI score of 0.9384, outperforming UFSPCA (0.9224) and other competing algorithms. This result reflects UFSOLPP's strength in preserving the intrinsic relationships among facial expressions in a nonlinear manifold space.
Yale Dataset: UFSOLPP obtained an NMI of 0.6363, slightly better than UFSPCA (0.6359) and closely matching RSR (0.6368). Given the Yale dataset's complexity, stemming from variations in lighting, pose, and expression, this

performance demonstrates the algorithm's robustness under challenging conditions.

ORL Dataset: With an NMI of 0.8356, UFSOLPP outperformed several methods, including SCEFS (0.7102) and AEFS (0.7031), highlighting its ability to identify discriminative features across classes with diverse facial structures.

COIL20 Dataset: UFSOLPP achieved an NMI of 0.9264, slightly below UFSPCA (0.9518), yet significantly better than SCEFS (0.4362) and DGUFS (0.5023). This indicates that UFSOLPP generalizes well to object image datasets, demonstrating flexibility beyond facial image domains.

Pixraw10P Dataset: On this high-dimensional dataset (10,000 features), UFSOLPP obtained an NMI of 0.8557, closely comparable to UFSPCA (0.8608) and superior to AEFS (0.8421) and SCEFS (0.7195). The result confirms the method's ability to extract relevant features despite the data's dimensional complexity.

Overall, UFSOLPP consistently achieved high NMI values across all datasets, confirming its effectiveness in preserving mutual information between selected features and class distributions. These results are particularly significant in datasets with nonlinear structures, such as facial image datasets with varying expressions and poses. The Wilcoxon test further validates that UFSOLPP significantly outperforms other methods in terms of NMI, reinforcing its reliability and generalizability. The best results in Table 6 are highlighted in bold.

The UFSPCA algorithm is fundamentally based on the classical Principal Component Analysis (PCA), which aims to reduce data dimensionality by preserving the maximum variance within the dataset. In essence, PCA extracts orthogonal principal components that represent the directions of highest variance. However, this process generally overlooks the local and neighborhood structure of the data in the original feature space. This limitation becomes critical when dealing with complex, nonlinear data manifolds such as face images, where preserving intrinsic geometric and local relationships is essential for effective feature representation.

In contrast, the Orthogonal Locality Preserving Projections (OLPP) algorithm extends the advantages of PCA by incorporating the preservation of local data structures. OLPP first applies PCA to project the data into a new space and then constructs an adjacency graph to model the local neighborhood relationships among samples. By computing the graph Laplacian and solving a corresponding eigenvalue problem, OLPP generates projections that preserve the locality structure of the data in the new feature space. This approach enables OLPP to maintain not only global variance but also important local geometric features, which are often lost in PCA alone. The proposed UFSOLPP method effectively combines these two algorithms, leveraging PCA's ability to extract orthogonal features while simultaneously utilizing OLPP's capacity to preserve local manifold structure. This synergy allows UFSOLPP to select highly informative and non-redundant features that better capture the intrinsic geometry of the data. Empirically, this combination results in superior

feature selection performance, particularly in high-dimensional and complex datasets such as Yale and ORL, where UFSOLPP consistently outperforms UFSPCA. By ensuring that the selected features reflect both the global variance and local relationships, UFSOLPP enhances classification accuracy and robustness, demonstrating improved discriminative power and generalizability compared to methods relying solely on PCA. Based on the experimental results, the proposed UFSOLPP method consistently achieves either the best or second-best performance across five benchmark image datasets. These empirical findings clearly indicate the superiority of UFSOLPP in comparison to eight state of the art algorithms. In addition to enhancing classification accuracy, UFSOLPP demonstrates strong robustness in handling high-dimensional and nonlinear data distributions, thereby establishing its effectiveness as an unsupervised feature selection method in complex learning scenarios.

The results obtained from the proposed method demonstrate its strong capability in selecting effective features from high-dimensional data with complex nonlinear structures. These characteristics enable broad applicability in real-world problems, particularly in areas such as face recognition systems in security applications, image retrieval in multimedia databases, and analysis of biological and medical data, which often involve noisy and correlated features. Moreover, the ability to preserve local data structures and select distinctive features without relying on class labels enhances the method's utility in unsupervised scenarios, commonly encountered in practical applications. Therefore, the findings of this study hold significant theoretical and practical implications for the analysis of complex, high-dimensional datasets.

4.4. **Computational Complexity.** This section presents the computational complexity analysis of the proposed UFSOLPP algorithm. The total complexity has three core components: Orthogonal Locality Preserving Projections (OLPP) [31], Pearson Correlation Similarity, and the Hungarian Algorithm. The time complexity of each component is detailed below. The OLPP component involves an iterative optimization process, with a time complexity of $O\ (T \times m^2 \times d)$, where m denotes the number of data samples, d is the number of features, and T is the number of iterations until convergence. The Pearson Correlation Similarity component computes the similarity between all pairs of data points, resulting in a time complexity of $O\ (m^2 \times d)$ [9]. The Hungarian Algorithm is applied to solve the maximum weighted matching problem in a bipartite graph constructed over the feature space. This step incurs a computational cost of $O\ (d^3)$, where d is the number of features and corresponds to the dimension of the cost matrix [9]. The overall computational complexity of the proposed algorithm is obtained by aggregating the individual complexities of its constituent components, resulting in $O\ (T \times m^2 \times d) + O\ (m^2 \times d) + O\ (d^3)$. By combining terms and omitting lower-order contributions, the total complexity can be simplified to $O\ (T \times m^2 \times d + d^3)$.

4.5. **Running Time.** Table 7 presents the average execution times (in seconds) of the proposed UFSOLPP method and the comparative algorithms that achieved higher accuracy in the experimental evaluations across five benchmark datasets. The OLPP component in UFSOLPP operates on vectorized data, where original image matrices are transformed into high-dimensional vectors. This vectorization leads to a considerable increase in dimensionality, which subsequently raises the computational complexity and simulation time. Despite this, UFSOLPP demonstrates acceptable runtime efficiency on most datasets. For instance, on the Pixraw10P dataset, the execution time of UFSOLPP reaches 67,478 seconds; this can be attributed to the extremely high dimensionality of this dataset (10,000 features), which poses significant challenges to most algorithms. In summary, while UFSOLPP incurs higher computational costs due to its use of OLPP and maximum matching, it maintains a reasonable balance between execution time and selection accuracy, especially for moderately-sized datasets.

TABLE 7. Average execution time (in seconds) of the proposed method UFSOLPP

Dataset	UFSOLPP	UFSPCA	RSR	URAFS
Jaffe	4.45	2.75	2.04	4.16
Yale	21.03	10.03	5.65	17.20
ORL	23.54	8.96	6.74	22.58
COIL-20	26.30	8.58	11.71	20.76
Pixraw10P	67478	52550	2546.9	5918.6

4.6. **Limitations.** Although the proposed UFSOLPP algorithm demonstrates strong performance in selecting relevant features from high-dimensional and nonlinear data, it has certain limitations that should be acknowledged. One of the primary constraints is the computational complexity, particularly related to the Orthogonal Locality Preserving Projections (OLPP) component and the Hungarian algorithm used for maximum matching. Specifically, OLPP involves iterative optimization with a complexity dependent on both the number of samples and features, while the Hungarian algorithm exhibits cubic time complexity with respect to the number of features. As a result, when applied to very large datasets or extremely high-dimensional feature spaces, the algorithm may require substantial computational resources and longer processing times, which could limit its applicability in real-time or resource-constrained environments. Future research could investigate optimization strategies or approximation techniques to reduce computational demands and improve scalability, thereby enhancing its applicability across a broader range of scenarios. Despite these limitations, the algorithm's ability to extract orthogonal features, preserve local structures, and effectively select discriminative features without

relying on labeled data provides significant benefits, especially in complex unsupervised learning scenarios.

5. Conclusion

This paper presented the UFSOLPP algorithm, a novel unsupervised feature selection method that incorporates Orthogonal Locality Preserving Projection (OLPP) to address the challenges posed by high-dimensional data, particularly in tasks such as face image classification. By constructing a nonlinear manifold of the data and extracting orthogonal features that preserve local structures, UFSOLPP effectively reduces noise and redundancy while enhancing the discriminative power of the selected features. Furthermore, the incorporation of a Weighted Bipartite Graph Matching (WBGM) strategy facilitates the selection of features that best represent the original data, thereby improving classification accuracy. The results demonstrate the algorithm's capability to efficiently eliminate redundant and noisy features while retaining essential information that is critical for achieving high classification performance. This capability is particularly important for datasets characterized by complex relationships and variations, such as facial images, where preserving nonlinear structures is essential. The versatility of the UFSOLPP algorithm allows its adaptation to semi-supervised and supervised feature selection tasks by integrating alternative feature extraction techniques such as Linear Discriminant Analysis (LDA) or Two-Dimensional Linear Discriminant Analysis (2DLDA). Experimental results on benchmark datasets demonstrated the superior performance of the UFSOLPP algorithm compared to other unsupervised feature selection methods. Future work may explore the use of evolutionary algorithms as alternatives to the Hungarian algorithm to further enhance feature selection efficiency. Additionally, optimization of computational procedures to reduce overall simulation time will be pursued. Such advancements would strengthen UFSOLPP's position as a robust and flexible tool for a broad spectrum of high-dimensional data analysis and classification tasks.

6. Aknowledgement

I would like to thank the reviewers for their thoughtful comments and efforts towards improving my manuscript.

7. Data Availability Statement

The data generated during the study are subject to a data-sharing mandate and are available in public repositories. All datasets used in this study have been appropriately cited in the manuscript.

8. Ethical considerations

The author avoided data fabrication and falsification.

9. Funding

This research did not receive any specific grant from the public, commercial, or not-for-profit funding agencies.

10. Conflict of interest

The author declare no conflict of interest.

References

[1] Zhao, X., Wu, D., Nie, F., Member, S., & Yu, W. (2024). Nonlinear Locality-Preserving Projections With Dynamic Graph Learning. *IEEE Transactions on Neural Networks and Learning Systems, PP*, 1–12. https://doi.org/10.1109/TNNLS.2024.3408835.

[2] Yao, Y., Meng, H., Gao, Y., Long, Z., & Li, T. (2023). Linear dimensionality reduction method based on topological properties. *Information Sciences, 624*, 493–511. https://doi.org/10.1016/j.ins.2022.12.098.

[3] Fattahi, M., Moattar, M. H., & Forghani, Y. (2023). Locally alignment-based manifold learning for simultaneous feature selection and extraction in classification problems. *Knowledge-Based Systems, 259*, 110088. https://doi.org/10.1016/j.knosys.2022.110088.

[4] Gerber, S., Tasdizen, T., Fletcher, P. T., Joshi, S., Whitaker, R., & Adni, I. (2010). Manifold modeling for brain population analysis. *Medical Image Analysis, 14(5)*, 643–653. https://doi.org/10.1016/j.media.2010.05.0081.

[5] Ahadzadeh, B., Abdar, M., Safara, F., Khosravi, A., Menhaj, M. B., & Suganthan, P. N. (2023). SFE: A Simple, Fast, and Efficient Feature Selection Algorithm for High-Dimensional Data. *IEEE Transactions on Evolutionary Computation, 27(6)*, 1896–1911. https://doi.org/10.1109/TEVC.2023.3238420.

[6] Qin, Z., Chen, H., Mi, Y., Luo, C., Horng, S., & Li, T. (2024). Multi-label feature selection with adaptive graph learning and label information enhancement. *Knowledge-Based Systems, 285*(December 2023), 111363. https://doi.org/10.1016/j.knosys.2023.111363.

[7] Lee, K. K. G., Kasim, H., Zhou, W. J., Sirigina, R. P., & Hung, G. G. T. (2023). Feature redundancy assessment framework for subject matter experts. *Engineering Applications of Artificial Intelligence, 117*(September 2022), 105456. `https://doi.org/10.1016/j.engappai.2022.105456`.

[8] Afshar, M., & Usefi, H. (2022). Optimizing feature selection methods by removing irrelevant features using sparse least squares. *Expert Systems with Applications, 200*(April), 116928. https://doi.org/10.1016/j.eswa.2022.116928.

[9] Beiranvand, F., Mehrdad, V., & Dowlatshahi, M. B. (2022). Unsupervised feature selection for image classification: A bipartite matching-based principal component analysis approach. *Knowledge-Based Systems, 250*, 109085. `https://doi.org/10.1016/j.knosys.2022.109085`.

[10] Xiang, L., Chen, H., Yin, T., Horng, S. J., & Li, T. (2024). Unsupervised feature selection based on a bipartite graph and low-redundant regularization. *Knowledge-Based Systems, 302*(June), 112379. https://doi.org/10.1016/j.knosys.2024.112379.

[11] Paniri, M., Dowlatshahi, M. B., & Nezamabadi-pour, H. (2020). MLACO: A multi-label feature selection algorithm based on ant colony optimization. *Knowledge-Based Systems, 192*, 105285. https://doi.org/10.1016/j.knosys.2019.105285.

[12] Hashemi, A., Dowlatshahi, M. B., & Nezamabadi-pour, H. (2021). An efficient Pareto-based feature selection algorithm for multi-label classification. *Information Sciences, 581*, 428–447. https://doi.org/10.1016/j.ins.2021.09.052.

[13] Ang, J. C., Mirzal, A., Haron, H., & Hamed, H. N. A. (2016). Supervised, unsupervised, and semi-supervised feature selection: A review on gene selection.

IEEE/ACM Transactions on Computational Biology and Bioinformatics, 13(5), 971–989. https://doi.org/10.1109/TCBB.2015.2478454.

[14] Hindawi, M., Elghazel, H., & Benabdeslem, K. (2013). Efficient semi-supervised feature selection by an ensemble approach. *Undefined*.

[15] Yao, C., Liu, Y. F., Jiang, B., Hen, J., & Han, J. (2017). LLE Score: A New Filter-Based Unsupervised Feature Selection Method Based on Nonlinear Manifold Embedding and Its Application to Image Recognition. *IEEE Transactions on Image Processing, 26(11)*, 5257–5269. https://doi.org/10.1109/TIP.2017.2733200.

[16] Xiao, Z., Chen, H., Mi, Y., Luo, C., Horng, S. J., & Li, T. (2025). Joint subspace learning and subspace clustering based unsupervised feature selection. *Neurocomputing, 635*(March), 129885. https://doi.org/10.1016/j.neucom.2025.129885.

[17] Wang, Y., Huang, M., Zhou, L., Che, H., & Jiang, B. (2024). Multi-cluster nonlinear unsupervised feature selection via joint manifold learning and generalized Lasso. 255(February).

[18] Li, D., Chen, H., Mi, Y., & Luo, C. (2024). Unsupervised feature selection via dual space-based low redundancy scores and extended OLSDA. 662(December 2023).

[19] Xiang, L., Chen, H., Yin, T., Horng, S. J., & Li, T. (2024). Unsupervised feature selection based on bipartite graph and low-redundant regularization. *Knowledge-Based Systems, 302*(February), 112379. https://doi.org/10.1016/j.knosys.2024.112379.

[20] Han, K., Wang, Y., Zhang, C., Li, C., & Xu, C. (2018). Autoencoder Inspired Unsupervised Feature Selection. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, 2018-April*, 2941–2945. https://doi.org/10.1109/ICASSP.2018.8462261.

[21] Guo, J., & Zhu, W. (2018). Dependence guided unsupervised feature selection. *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, 2232–2239. https://doi.org/10.1609/aaai.v32i1.11904.

[22] Selection Algorithms Based on Standard Deviation and Cosine Similarity for Genomic Data Analysis. *Frontiers in Genetics, 12*. https://doi.org/10.3389/fgene.2021.684100.

[23] Zhu, P., Zuo, W., Zhang, L., Hu, Q., & Shiu, S. C. K. (2015). Unsupervised feature selection by regularized self-representation. *Pattern Recognition, 48(2)*, 438–446. https://doi.org/10.1016/j.patcog.2014.08.006.

[24] Huang, D., Cai, X., & Wang, C. D. (2019). Unsupervised feature selection with multi-subspace randomization and collaboration. *Knowledge-Based Systems, 182*(July). https://doi.org/10.1016/j.knosys.2019.07.027.

[25] Li, X., Zhang, H., Zhang, R., Liu, Y., & Nie, F. (2019). Generalized Uncorrelated Regression with Adaptive Graph for Unsupervised Feature Selection. *IEEE Transactions on Neural Networks and Learning Systems, 30(5)*, 1587–1595. https://doi.org/10.1109/TNNLS.2018.2868847.

[26] Samareh-Jahani, M., Saberi-Movahed, F., Eftekhari, M., Aghamollaei, G., & Tiwari, P. (2024). Low-Redundant Unsupervised Feature Selection based on Data Structure Learning and Feature Orthogonalization. *Expert Systems with Applications, 240*(October 2023), 122556. https://doi.org/10.1016/j.eswa.2023.122556.

[27] Xiao, Z., Chen, H., Mi, Y., Luo, C., Horng, S. J., & Li, T. (2025). Joint subspace learning and subspace clustering based unsupervised feature selection. *Neurocomputing, 635*(March), 129885. https://doi.org/10.1016/j.neucom.2025.129885.

[28] Ma, Z., Wei, Y., Huang, Y., & Wang, J. (2024). Unsupervised feature selection based on minimum-redundant subspace learning with self-weighted adaptive graph. *Digital Signal Processing: A Review Journal, 155*(August), 104738. https://doi.org/10.1016/j.dsp.2024.104738.

[29] Zhang, C., Nie, F., Wang, R., & Li, X. (2024). Supervised Feature Selection via Multi-Center and Local Structure Learning. *IEEE Transactions on Knowledge and Data Engineering, 36(9)*, 4930–4942. https://doi.org/10.1109/TKDE.2024.3372657.

[30] Zohrati, L., Abadeh, M. N., & Kazemi, E. (2018). Flexible approach to schedule tasks in cloud-computing environments. *IET Software, 12(6)*, 474–479. https://doi.org/10.1049/iet-sen.2017.0008.

[31] Cai, D., He, X., Han, J., & Member, S. (2006). Orthogonal Laplacianfaces for Face Recognition. 15(11), 3608–3614.

[32] Liu, B. Di, Wang, Y. X., Zhang, Y. J., & Shen, B. (2013). Learning dictionary on manifolds for image classification. *Pattern Recognition, 46(7)*, 1879–1890. https://doi.org/10.1016/j.patcog.2012.11.018.

[33] Ding-cheng, F. (n.d.). Detecting Local Manifold Structure for Unsupervised Feature Selection. *Acta Automatica Sinica, 40(10)*, 2253–2261. https://doi.org/10.1016/S1874-1029(14)60362-1.

[34] Yu, J., & Zhang, C. (2020). Manifold regularized stacked autoencoders-based feature learning for fault detection in industrial processes. *Journal of Process Control, 92*, 119–136. https://doi.org/10.1016/j.jprocont.2020.06.001.

[35] Yousuff, M., Babu, R., & Rathinam, A. (2024). Nonlinear dimensionality reduction based visualization of single-cell RNA sequencing data. *Journal of Analytical Science and Technology, 15(1)*, 1–23. https://doi.org/10.1186/s40543-023-00414-0.

[36] Belkin, M., & Niyogi, P. (2002). Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in Neural Information Processing Systems*. https://doi.org/10.7551/mitpress/1120.003.0080.

[37] Kienitz, D., Komendantskaya, E., & Lones, M. (2022). The Effect of Manifold Entanglement and Intrinsic Dimensionality on Learning. *Proceedings of the AAAI Conference on Artificial Intelligence, 36(7)*, 7160–7167. https://doi.org/10.1609/aaai.v36i7.20676.

[38] Question, S. (2012). Pearson's correlation coefficient. 4483(July), 1–2. https://doi.org/10.1136/bmj.e4483.

[39] Armiti, A., & Gertz, M. (n.d.). Geometric Graph Matching and Similarity: A Probabilistic Approach. Categories and Subject Descriptors.

[40] MANUAL, SOLUTION. *Introduction to Graph Theory*.

[41] Schrijver, A. (2017). *A Course in Combinatorial Optimization*.

[42] Dwivedi, S. P. (n.d.). Approximate Bipartite Graph Matching. Springer Singapore. https://doi.org/10.1007/978-981-16-8546-0.

[43] Schwartz, J., Steger, A., & Weißl, A. (2005). Fast Algorithms for Weighted Bipartite Matching, 476–487.

[44] Rahimi, Z., Taghipour, K., Khadivi, S., & Afhami, N. (2012). Document and sentence alignment in comparable corpora using bipartite graph matching. In *2012 6th International Symposium on Telecommunications (IST)*, November, 817–821. https://doi.org/10.1109/ISTEL.2012.6483098.

[45] Zhu, H., Liu, D., Zhang, S., Zhu, Y., & Teng, L. (2016). Solving the Many to Many assignment problem by improving the Kuhn–Munkres algorithm with backtracking. 618, 30–41. https://doi.org/10.1016/j.tcs.2016.01.002.

[46] Beiranvand, F., Mehrdad, V., & Dowlatshahi, M. B. (2025). Unsupervised feature selection based on the two-dimensional principal component analysis and bipartite graph for face image classification. *Journal of Mahani Mathematical Research*, February 1.

[47] Tan, H., Zhang, X., Guan, N., Tao, D., & Huang, X. (2015). Two-Dimensional Euler PCA, 548–559.

[48] Nene, S. A., Nayar, S. K., & Murase, H. (1996). Columbia Object Image Library (COIL-20), Tech. Rep. CUCS-005-96, Dept. Comput. Sci., Columbia Univ., New York, NY, USA.

[49] Arafat, H., Alfeilat, A., Hassanat, A. B. A., Lasassmeh, O., & Tarawneh, A. S. (2019). Effects of Distance Measure Choice on K-Nearest Neighbor Classifier Performance: A Review, 00(00). https://doi.org/10.1089/big.2018.0175.

[50] Liao, Y., & Vemuri, V. R. (2002). Use of k-nearest neighbor classifier for intrusion detection. *Computers & Security*, 21(5), 439–448.
[51] Jain, D., & Singh, V. (2018). Feature selection and classification systems for chronic disease prediction: A review. *Egyptian Informatics Journal*, 19(3), 179–189. https://doi.org/10.1016/j.eij.2018.03.002.

Firoozeh Beiranvand
Orcid number: 0000-0002-1908-8386
Department of Electrical Engineering
Lorestan University
Khoramabad, Iran
Email address: beiranvand.fi@fe.lu.ac.ir