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

UNSUPERVISED FEATURE SELECTION BASED ON THE TWO-DIMENSIONAL PRINCIPAL COMPONENT ANALYSIS AND BIPARTITE GRAPH FOR FACE IMAGE CLASSIFICATION

F. Beiranvand, V. Mehrdad ✉, and MB. Dowlatshahi

Article type: Research Article

(Received: 11 June 2024, Received in revised form 22 September 2024)

(Accepted: 17 January 2025, Published Online: 01 February 2025)

Abstract. In this paper, we propose a new matrix-based feature selection method, called UFS2DPCA, which leverages the hidden knowledge in orthogonal features obtained from two-dimensional principal component analysis (2DPCA) to perform accurate unsupervised feature selection. The UFS2DPCA algorithm first uses 2DPCA to directly extract uncorrelated and orthogonal features from the two-dimensional image datasets. We then compute the correlation similarity between the main and extracted features. Finally, a weighted bipartite graph is constructed using two sets of features, and the best features are selected using the fast LAPJV algorithm. The selected features are classified using the K-Nearest Neighbor (KNN) classifier. To ensure statistical significance, the Friedman test is applied to compare the performance of UFS2DPCA with other methods. The algorithm is evaluated on four well-known image datasets: Jaffe, Yale, ORL, and pixraw10P. Key performance metrics such as accuracy, normalized mutual information (NMI), precision, recall, and F-measure are used for evaluation. The experimental results show that UFS2DPCA consistently outperforms other state-of-the-art unsupervised feature selection methods. For example, UFS2DPCA achieves an average NMI of 0.9244 and an average accuracy of 0.9033 on the pixraw10P face image dataset that has 10000 features. Similarly, it demonstrates superior performance in accuracy, recall, Precision, F-measure, and NMI across all datasets.

Keywords: Two-Dimensional Principal Component Analysis, Weighted Bipartite Graph Matching, LAPJV algorithm, Augmenting Path, similarity.
2020 MSC: 68T20.

1. Introduction

Technological advancements have led to the generation of vast amounts of complex data. However, the high dimensionality of this data introduces challenges such as increased computational complexity, increased memory usage,

✉ mehrdad.v@lu.ac.ir, ORCID: 0000-0003-2121-9119
https://doi.org/10.22103/jmmr.2025.23505.1659
Publisher: Shahid Bahonar University of Kerman
How to cite: F. Beiranvand, V. Mehrdad, MB. Dowlatshahi, *Unsupervised feature selection based on the two-dimensional principal component analysis and bipartite graph for face image classification*, J. Mahani Math. Res. 2026; 15(1): 1-37.

and reduced algorithm efficiency due to the presence of irrelevant and redundant features (Wahid et al., 2022). Feature extraction and feature selection are two approaches to solving this problem. Feature extraction is a conventional technique that transforms the input space into a subspace with lower dimensions while keeping maximum relevant information. Feature extraction reduces complexity by displaying each feature as a linear combination of the original features in the feature space (Khalid, Khalil, & Nasreen, 2014). A way to select the best features from given features is feature selection (Ahamad & Ahmad, 2021). Feature selection is applicable in data mining and machine learning, including applications in robotics, engineering, pattern recognition (handwriting, speech, and face recognition), econometric and marketing, internet (text categorization), and medical applications (medicine (discovery), Prognosis, and diagnosis). Feature selection has many economic advantages such as reduced computational and memory requirements, and data collection costs. It also improves understanding of the dataset and prediction process. Feature selection involves making good predictions with the least features possible. Feature selection is needed to "overcome the curse of dimensionality"(Guyon, 2008).

Feature selection based on the search is categorized into three methods, i.e., embedded, wrapper, and filter approach. Filter methods select distinct features by using the inherent characteristics of the data (Miao & Niu, 2016). The filter approach relies upon the general properties of the training data, such as distance, information, dependence, and consistency, and does not use a learning algorithm (Yu & Liu, 2004). They use the statistical characteristics of the data to determine the distinctive features. Therefore, they are very beneficial, effective, and a favorite for the high-dimensionality dataset (Paniri, Dowlatshahi, & Nezamabadi-pour, 2021). The filter methods are divided into two categories univariate and multivariate. If only the importance of each feature is studied separately and its correlation with other features is not considered, they belong to the univariate category, and if the importance between features is considered, they belong to the multivariate category (Beiranvand, Mehrdad, & Dowlatshahi, 2022). Thus, the efficiency of the multivariate methods outperforms the univariate methods, but in terms of computational cost, it is more expensive than the univariate methods (Tabakhi, Moradi, & Akhlaghian, 2014). Wrapper methods require the use of learning algorithms to evaluate features (Miao & Niu, 2016) and determine the best subsets of selected features. Thus, they have a high computational cost for data with a high number of features (Yu & Liu, 2004). Wrapper methods offer better feature subsets than filter methods but require massive amounts of computation and are also time-consuming because of the use of learning algorithms and therefore not appropriate for high-dimensional datasets (Paniri, Dowlatshahi, & Nezamabadi-pour, 2021). Embedded methods select the feature in the model construction process (Miao & Niu, 2016). The embedded model includes a cost function and a regularization term. The term related to the feature is constructed to relate the features in the cost function, and the sparse regularization term is applied

to rank the features in this term (Li et al., 2022). Furthermore, according to class labels, the feature selection technique has three categories: supervised, semi-supervised, and unsupervised methods. In supervised feature selection algorithms, the class labels are available, and relevant features are recognized to differentiate samples from different class labels. In semi-supervised feature selection, only a few portions of the data are labeled, and in this method, a similarity matrix is usually created and the features that most closely match this matrix are selected. In an unsupervised feature selection, the class labels aren't available, which is a hard problem (Miao & Niu, 2016). Feature selection is an important task in image classification. In image processing, such as face recognition and pattern recognition, data is usually in the form of a matrix. An image is stored as a matrix, and if we want to use vector-based methods, we must convert it to a vector, which causes the loss of the structure and relation in the original data matrix. In addition, a vector with high dimensionality is produced, which causes highly complex calculations and singularity problems. To solve this problem, the original data matrix is used for learning (Yuan, Li, Lai, & Tang, 2020). An image describes many features, but few of them are effective and efficient for classification (Zhou & Wang, 2015).

A major challenge in high-dimensional data is the correlation between features, which results in information redundancy. Redundant information reduces the performance of machine learning algorithms by introducing noise and over-complicating the model. While various feature selection methods exist, most fail to effectively handle high-dimensional image data. Traditional approaches such as Principal Component Analysis (PCA) transformed matrices into vectors, causing the loss of important spatial relationships within the data. These one-dimensional methods often overlooked critical inter-feature dependencies, leading to suboptimal feature selection, especially in cases where the data's original structure is important. This research aims to address the following question:

• How can we select features that closely resemble the orthogonal and uncorrelated features extracted by the 2DPCA algorithm, while preserving the structure of the images and reducing the dimensionality of the feature vectors? On the other hand, one of the significant challenges in unsupervised feature selection is the lack of class labels, which makes it difficult to determine the relevance of features based on output variables. Unlike supervised methods that utilize labeled data to guide feature selection, unsupervised methods must rely solely on the internal structure of the data, such as feature correlations or intrinsic patterns. This often leads to the challenge of identifying the most informative and diverse features without overfitting or losing important information. Despite these challenges, the proposed algorithm is designed to perform unsupervised feature selection with high accuracy. By focusing on features that are orthogonal and uncorrelated, similar to those obtained through the 2DPCA algorithm, the method can effectively capture the underlying structure of the data without relying on labels. This results in a more accurate and efficient

selection of features, even in complex, high-dimensional datasets where class information is unavailable. It should be noted that no learning algorithms are used in this proposed method and it belongs to the multivariate filter category.

The major contributions of this study are summarized below:

1. We directly extract the orthogonal and uncorrelated features from the two- dimensional image dataset using the 2DPCA feature extraction approach. Feature extraction takes the image as input and extracts its features in the output. Each image is stored as a matrix. Pixels play the role of matrix elements. Each pixel is a raw feature. The new features extracted using 2DPCA are called principal components (PC). In this step, we do not reduce the dimensionality. We only transform the original features (X) to a new space and obtain new independent features (Y). The remarkable point is that the dimensions of Y and X are the same. 2. We compute the similarity between the original features of the two-dimensional dataset of images and all extracted features using linear correlation similarity and store the results in a matrix called matrix C.

3. Now, using three sets including original features, extracted features, and a similarity matrix we create a weighted bipartite graph matching. This bipartite graph matching does not have any loops. We use the LAPJV algorithm to find maximum matching. The best features are the subset of original features placed in maximum matching.

The continuation of our paper is planned as follows: In the second Section, we summarize and describe some proposed feature selection methods. We introduce our UFS2DPCA proposed method and the theoretical concepts used in Section 3. In section 4, we show the experimental results. Finally, Section 5 explains future works and concludes.

2. **Related work**

In the field of feature selection, many works have been done, and we will review some of the most famous ones. Li et al. (2022) proposed a new two-dimensional weight-based unsupervised feature selection (2DWFS). In this method, a filter was designed to learn the feature weight ranking as a substitute for using sparse regularization. Then, they proposed two methods 2DWFS-1 and 2DWFS-2, which had two main parameters λ_1 and λ_2 (Li et al., 2022). Xiang et al. (2024) proposed a novel unsupervised feature selection method, BGLR, which selected anchors based on variance and built an adaptive bipartite graph in the projection space. Their approach incorporated a low-redundancy regularization term to control feature redundancy and improve efficiency in large-scale data processing (Xiang et al., 2024). Yuan et al. (2020) proposed a two-dimensional semi-supervised feature selection (2DSSFS). 2DSSFS combined the sparse matrix regression method and the label prediction and constructed an optimization model to obtain pseudo labels of the unlabeled data (Li, Liang, & Li, 2020). Karami et al. (2023) introduced the Variance–Covariance subspace distance, which effectively extracted variance

and covariance information from dataset features. They established an unsupervised feature selection method, VCSDFS, combining this distance with inner product regularization to facilitate feature selection while minimizing redundancy and achieving effective subspace learning (Karami,Saberi-movahed, Tiwari, & Marttinen, 2023). Wang et al. (2023) introduced the LLSRFS framework for unsupervised feature selection, which directly learned feature weights and incorporated an exponential weighting mechanism to reduce the dominance of large-weight features. Their approach effectively preserved local graph structures and demonstrated significant improvements over existing methods (Wang, Wang, Gu, Wei, & Liu, 2024). Han et al. (2018) proposed an auto-encoder-based feature selection technique called AEFS, they compressed input features using an auto-encoder network in low-dimensional space, and features that had little effect on low-dimensional space data were deemed redundant and were removed by the group sparsity regularization term (Han, Wang, Zhang, Li, & Xu, 2018). Guo et al. (2018) proposed a projection-free feature selection model called DGUFS (Guo & Zhu, 2018). Xie et al. (2021) proposed an SCFS algorithm for feature selection, which included the SCEFS and SCAFS algorithms (Xie, Wang, Xu, Huang, & Grant, 2021). Zhu et al. (2015) proposed an RSR model, where they used the $l_{2,1}$ norm to formalize and measure the presentation coefficients (Zhu, Zuo, Zhang, Hu, & Shiu, 2015). Huang et al. (2019) proposed a model named SRCFS for unsupervised feature selection, arguing that incremental random subspaces could improve unsupervised feature selection accuracy (Huang, Cai, & Wang, 2019). Li et al. (2018) proposed the URAFS algorithm, which used the maximum adaptive graph structure (Li, Zhang, Zhang, Liu, & Nie, 2019). Huang et al. (2022) introduced the AGDS method for unsupervised feature selection, which adaptively assigned weights to the k-nearest neighbors to address imbalanced neighbor issues. Their approach also incorporated a dependency score using mutual information and entropy to effectively eliminate redundant features (Huang & Yang, 2022). Hashemi et al. (2021) modeled a Multi-Criteria Decision-Making (MCDM) procedure for feature selection and used the VIKOR method (Hashemi, Dowlatshahi, & Nezamabadi-pour, 2021). Furthermore, Hashemi et al. (2021) proposed feature selection for multi-label data (Hashemi, Bagher, & Nezamabadi-pour, 2021a, Hashemi, Bagher, & Nezamabadi-pour, 2020, Hashemi, Bagher, & Nezamabadi-pour, 2021b), and also proposed FLG, which built a bipartite graph matching where binary class labels and the features were parts of the graph (Hashemi, Dowlatshahi, & Nezamabadi-Pour, 2021). Beiranvand et al.(2022) proposed the UFSPCA algorithm (Beiranvand, Mehrdad, & Dowlatshahi, 2022), which transformed the original feature vector dataset into a new space using Principal Component Analysis (PCA), obtaining uncorrelated and orthogonal features. The Hungarian algorithm was employed to select the best features.

In this paper, we utilized the 2DPCA method to directly extract uncorrelated features from 2D images and selected the most important ones using

the LAPJV algorithm. By incorporating the LAPJV algorithm, we achieved faster simulation times, and the direct feature extraction from 2D images via 2DPCA enhanced the simulation accuracy. It is important to note that PCA operated on one-dimensional vectors, requiring two-dimensional matrices to be reshaped into vectors. This process resulted in high-dimensional image vectors, making it challenging to accurately compute the covariance matrix. In contrast, 2DPCA worked directly on two-dimensional image matrices without the need for vectorization during the feature extraction step (Yang et al., 2004, Yang, J., & Liu, C., 2007). Furthermore, 2DPCA directly calculated the image covariance matrix from the original image matrices. This approach offered the advantage of producing a smaller covariance matrix compared to PCA, enabling more accurate covariance matrix evaluation and reducing the time needed to compute the required eigenvectors (Yang et al., 2004). Additionally, each principal component (PC) in 2DPCA was a vector, whereas in PCA, each PC was a scalar. Current two-dimensional unsupervised feature selection algorithms typically use sparse regularization terms to highlight important features, which increases the number of hyperparameters. Also, they still could not, in theory, express the relation between the sparse regularization term and feature selection. Almost all of the methods that we reviewed, such as SOGFS, RSR, DGUFS, AEFS, URAFS, and SRCFS, used hyperparameters, and their computation cost was very expensive. The advantage of the proposed UFS2DPCA algorithm was its ability to directly extract uncorrelated features from two-dimensional images and represent features using 2DPCA and bipartite graph matching. Our proposed UFS2DPCA algorithm did not use any hyperparameters or learning algorithms.

3. The Proposed UFS2DPCA Method

In many image datasets, the correlation among features can lead to problems in machine learning processes, as this correlation increases information redundancy and decreases model accuracy. Therefore, we decided to combine features in a new space to create orthogonal features that operate independently of one another. In this regard, we propose a new algorithm that helps select features that are very similar to orthogonal features. To extract these orthogonal features, we utilize the 2DPCA method, which is capable of extracting uncorrelated and intrinsic features from image data. This allows us to obtain features that are not only independent of one another but also reveal significant information contained in the data. To model the feature selection problem, we construct a weighted bipartite graph in which each feature is represented as a vertex and the similarity between features is represented as edge weights. The edge weights are calculated based on Pearson correlation, and then we use the LAPJV algorithm to find the maximum matching in this weighted bipartite graph. The initial features in this maximum matching are the selected features that are very similar to the orthogonal features. This approach enables

us to select features with minimal redundancy and the highest quality while preserving the structure of the data. This study aims to develop and evaluate a novel method for selecting features that are similar to orthogonal and uncorrelated features obtained using the 2DPCA method, utilizing weighted bipartite graph matching. We used three techniques consecutively. Step one is to use the 2DPCA technique to directly extract orthogonal and uncorrelated features from the two-dimensional image dataset. Step two is to calculate the pairwise similarity between the original features of the two-dimensional image dataset and the orthogonal extracted feature matrix by Pearson correlation. In the last step, we construct a graph that has two parts: the original features and the extracted feature sets, where the similarity matrix serves as the weights of edges between them. Each feature is represented by a vertex. Finally, we apply the LAPJV algorithm to obtain a matching with the largest size. We select the initial features in this maximum matching as the final features. Then we sort their index incrementally and measure the testing set with these features using the KNN classifier. Figure 1 illustrates the flowchart of the proposed UFS2DPCA method in full detail.

3.1. **Formulation.** In this section, we explain the three basic phases of the suggested UFS2DPCA method. Two-dimensional datasets include gray-level images (Lu et al., 2011). Each image is represented as a matrix. One of the significant characteristics of matrix data is the correlation between rows and columns, which, through vectorization, results in the loss of structural information. Vectorization also converts matrix data into high-dimensional vectors that require complex computations and large storage spaces. For example, if the dimensions of an image are 32×32, the size of the vector is 1024 (Li, Liang, & Li, 2020). Feature extraction is a method of transforming raw data into numerical features. In image recognition, subspace learning methods such as 2DPCA are used to extract image features (Wang & Gao, 2016). The 2DPCA method (Yang et al., 2004) is an unsupervised projection approach that ignores class labels. The image covariance matrix is directly calculated on the image matrices, thus preserving construction details. The covariance matrix of the image can be calculated with high accuracy, and it is usually full rank. As a result, it avoids dimensionality and small sample size (SSS) problems (Yang et al., 2004, Yang & Liu, 2007, Sanguansat, 2012). Covariance calculates the linear correlation between two random features, allowing us to determine whether there is a relationship between the two datasets.
The covariance matrix provides essential information about the dataset. For example, values close to zero represent uncorrelated features effective for classification, while very high or low values represent correlated features that provide no new information for identifying groups in the data (Nixon, 2007).

3.1.1. *2DPCA Algorithm.* The 2DPCA method was applied for the recognition of gray-level images (Wang et al., 2008) and proved to be a useful method for image feature extraction based on a two-dimensional image matrix (Zhang et

al., 2015). First, we computed the covariance matrix of the two-dimensional images, then found all its eigenvalues and eigenvectors using eigenvalue decomposition. We extracted the features of the images by representing the original image matrix in the feature subspace (Wang et al., 2021). The details of the implementation steps of the 2DPCA algorithm are as follows. Let an image be an $m \times n$ matrix P and X be an n-dimensional unitary column vector. Projection P into X by the image projection technique shown in equation (1) is the main idea (Yang et al., 2004):

$$Y=PX \tag{1}$$

The optimal projection vector X is obtained from the covariance matrix (S_x) by equation (2) (Kong et al., 2005):

$$S_x = E\left[(P - E(P))X\right]\left[(P - E(P))X\right]^T \tag{2}$$

The trace of S_x is denoted by equation (3):

$$\mathrm{tr}(S_x) = X^T E\left\{[P - E(P)][P - E(P)]^T\right\} X \tag{3}$$

$E\ (P)$ denotes the expectation, and transpose is marked with the letter T. The image covariance matrix of the training sample denoted by $\mathrm{G_t}$, is calculated for $i = (1, 2, 3\ldots\ M)$ by equations 4 (Kong et al., 2005):

$$G_t = E\left\{[P - E(P)]\ [P - E(P)]^T\right\} \tag{4}$$

Also, $\mathrm{G_t}$ can be directly calculated by the training image samples. If M denotes the number of training image samples and the ith training image samples is an $m \times n$ matrix P_i (i=1, 2, ..., M), then $\mathrm{G_t}$ is computed by equation (5) (Yang et al., 2004)):

$$G_t = \frac{1}{M}\sum_{i=1}^{M}(P_i - \overline{P})^T(P_i - \overline{P}) \tag{5}$$

G_t is a matrix with dimensionality $n \times n$. $\overline{P}$ shows the average image of all training samples (Yang et al., 2004). The eigenvectors obtained from G_t represent the direction of the optimal projection. We use all the eigenvectors derived from G_t. Then all the projected feature vectors Y are calculated from the transform $Y = P\ X_d,\ d = (1, 2, ..., n)$. Finally, the feature matrix Y is formed as $Y = [Y_1,\ Y_2, \ldots,\ Y_d]$ (Kong et al., 2005). It is a unitary vector by calculating the eigenvectors. The 2DPCA algorithm is shown in Algorithm 1.

In step 1 of the proposed method, our goal is to find all orthogonal and uncorrelated features. If the selected features are not orthogonal, we can say that the selected features are not good, so they should be removed and another feature should be selected. But when the features are orthogonal, we say with confidence that they are some of the important and effective features for classification and should not be removed. Therefore, the 2DPCA algorithm fulfills our goal of feature selection by constructing such features. We argue that the original features that are similar to orthogonal features obtained from

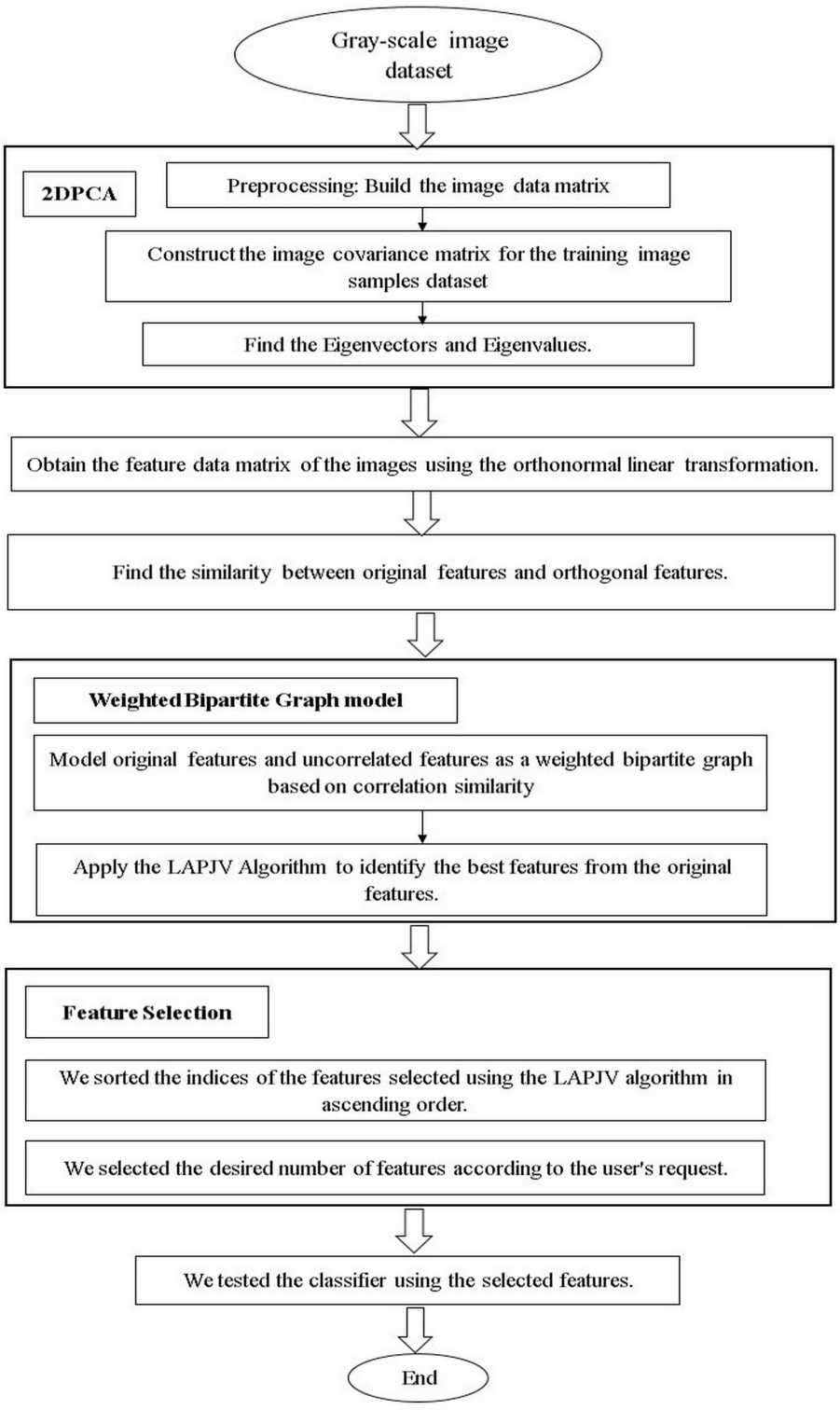


FIGURE 1. The general flowchart of UFS2DPCA.

2DPCA are non-removable and important features that should only be used in the classification.

3.1.2. *Correlation Similarity.* In step 2, we decide to calculate the pairwise distance between two sets of original features (X) and extracted features (Y) using the Pearson correlation method and we reach favorable results. Pearson's correlation coefficient is calculated based on mean and variance; we can use it since our data has a variance. One of the characteristics of Pearson's correlation coefficient is that it does not depend on the data measurement unit. The number of features of the two sets, Y and X, is equal. For the two sets (X, Y), the linear Pearson correlation coefficient PCC is given by:

$$PCC = \frac{\Sigma\ (Y - \overline{Y})(X - \overline{X})}{\sqrt{\Sigma\ (Y - \overline{Y})^2 (X - \overline{X})^2}} \tag{6}$$

$\overline{Y}$ and $\overline{X}$ denote the average of Y, and X, respectively. PCC is a number between -1 and 1.

If PCC becomes -1 or 1, Y and X are totally correlated, if PCC becomes 0 it means Y and X are uncorrelated. The distance between two feature sets Y and X is obtained by equation number 7 (Beiranvand, Mehrdad, & Dowlatshahi, 2022):

$$\text{Correlation distance(dis)} = 1 - \text{PCC} \tag{7}$$

The result of the Correlation distance (dis) is a number between 0 and 2. After the calculations mentioned above, we obtain the correlation similarity from equation 8 (Beiranvand, Mehrdad, & Dowlatshahi, 2022):

$$\text{Correlation similarity}(C) = \frac{1}{2}(2 - \text{dis}) \tag{8}$$

The maximum similarity number is one, which is obtained for a $dis = 0$. We model the proposed method using a weighted bipartite graph, and equation 8 determines the weight of the edges.

3.1.3. *Weighted Bipartite Graph Matching.* We obtain the new uncorrelated and orthogonal features, we examine different models to model the problem, and the best model is the bipartite graph model we choose. We consider the original features and the features resulting from the 2DPCA feature extraction method as two parts of the graph that are the same size and consider each vertex as a feature. Then we use Pearson's correlation similarity to define the weights of the edges. Maximum matching finds the optimal correspondence between two sets X and Y, so we need to obtain the maximum matching. First, we explain the basic concepts of graph theory. A graph $G = (V, E)$ is bipartite if the vertex set V can be split into two disjoint subsets V_1 and V_2 so that every edge in E connects a vertex in V_1 and V_2. There are no loops in a bipartite graph (Vasudev, 2006). A matching M is a part of E where

any 2 edges have a joint vertex (Matching, 1995), that is, $e \cap e^{'} = 0$ for all $e, e^{'} \in M$and $e \neq e^{'}$. If a matching covers all vertices, it is called a perfect matching (Schrijver, 2012). If an edge $(u_1 v_1)$ is in a matching M, then the two vertices u_1 and v_1 are called matched, and the vertices that are not matched by M are called exposed (Matching, 1995). An alternating path concerning a matching M is a path whose edges are alternately elements of M and not (Derigs, 1981, Manual, n.d).

Algorithm 1 :2DPCA Algorithm

Input: The image database contains a set of M training image samples $P_i \in \mathbb{R}^{m \times n}$ $(i = 1, 2, \ldots, M)$. The image dimension $(m \times n)$ projected a two-dimensional matrix.

Output: $n \times n$ orthogonal and uncorrelated feature matrix $Y \in \mathbb{R}^{m \times n}$.

1: **1) Begin**

2: **2) Calculate the average of all training image samples $\overline{P}$:**

$$\overline{P} = \frac{1}{M} \sum_{i=1}^{M} P_i$$

3: **3) Subtract the mean matrix $\overline{P}$ from the original image matrix P_i.**

4: **4) Construct the image covariance (scatter) matrix G_t:**

$$G_t = \frac{1}{M} \sum_{i=1}^{M} (P_i - \overline{P})^T (P_i - \overline{P})$$

where T denotes the transpose.

5: **5) Determine the eigenvalue and eigenvector by eigenvalue decomposition:** the total scatter of the projected samples, $J(X) = X^T G_t X$.

6: **6) Calculate all the projected feature vectors Y from the transform:**

$$Y_{m \times n} = P X_d, \quad d = (1, 2, \ldots, n)$$

and form the feature matrix $Y = [Y_1, Y_2, \ldots, Y_d]$.

An augmenting path is an alternating path between two unmatched vertices (Derigs, 1981). The M-augmenting path is a basic concept in matching theory. A path $P = (p_0, p_1, \ldots, p_t)$ in G is M-augmenting if it meets the following three conditions (Schrijver, 2012):

(1) t is odd,
(2) $p_1 p_2, p_3 p_4, \ldots, p_{t\text{-}2} p_{t\text{-}1} \in M$,
(3) $p_0, p_t \notin \cup M$.

An alternating path P is M-augmenting if both its end vertices are exposed (Matching, 1995, Manual, n.d.). A matching that has a maximum size among all matching in a graph is called a maximum matching (Manual, n.d., Rahimi et

al., 2012). Finding a matching with maximum weight is known as the maximum weight matching problem, and augmenting paths help solve it (Matching, 1995, Schrijver, 2012). The cover of the vertex is a subset W of V for each edge e of G (Schrijver, 2012). König's matching theorem states that if $V(G) = \max|M|$, where M is matching, and $T(G) = \max|W|$, then $V(G) = T(G)$ (Schrijver, 2012). Let $w : E \to \mathbb{R}$ be a 'weight' function. The weight $w(M)$ of E is calculated by equation (9):

$$w(M) = \sum_{e \in M} w(e) \tag{9}$$

A matching that matches each vertex is a perfect matching. In this paper, we use the LAPJV algorithm to find maximum matching. An example of maximum matching bipartite graph is illustrated in Figure 2.

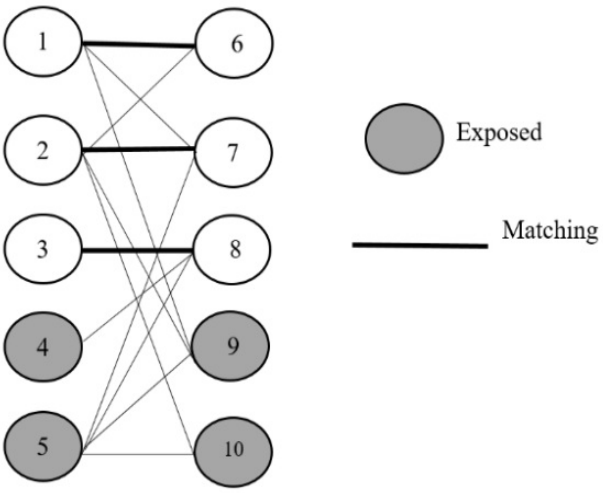


FIGURE 2. Maximum matching in a bipartite graph: (1; 9), (2; 6), (3; 8), and (5; 7) (Beiranvand, Mehrdad, & Dowlatshahi, 2022).

It is worth mentioning that the graph design is not simple for a large number of vertices, and we calculate it with a matrix. Next, we will explain the LAPJV algorithm (Jonker & Volgenant, 1988) in detail. The LAPJV algorithm, developed by Jonker and Volgenant (1988) (Jonker & Volgenant, 1988), is a highly efficient method for solving both dense and sparse linear assignment problems.It is classified as a shortest augmenting path algorithm, known for its superior efficiency compared to existing methods for such problems. This algorithm employs Dijkstra's shortest path technique and includes initialization steps that promote quicker convergence to optimal solutions (Jonker &

Volgenant, 1988, Jones et al., 2017). The LAPJV algorithm is divided into two primary phases: the Initialization Phase and the Augmentation Phase (Jonker & Volgenant, 1988, Jones et al., 2017). The Initialization Phase comprises three essential steps:

1. **Column Reduction:** The minimum value from each column in the cost matrix is subtracted from all elements in that column, ensuring that each column has at least one zero, which can be assigned to a corresponding row (Jonker & Volgenant, 1988, Jones et al., 2017).

2. **Reduction Transfer:** After column reduction, a positive value is subtracted from each row. This step maintains feasibility by ensuring that each assigned row has at least one zero, preventing negative values from arising in the matrix. The second smallest value in the row is then added to the column with the first zero found, balancing the reduction (Jonker & Volgenant, 1988, Jones et al., 2017).

3. **Augmenting Row Reduction**: This step seeks alternate paths from unassigned rows to unassigned columns. The algorithm identifies the minimum and second minimum values for each unassigned row and reduces the row by the second minimum value. If this results in a negative entry, a reverse column reduction is applied to keep the solution feasible (Jonker & Volgenant, 1988, Jones et al., 2017).

Following initialization, the augmentation phase focuses on constructing a matching and identifying the shortest augmenting path between unassigned rows and assigned columns. This iterative process updates assignments to minimize the overall cost in the matrix. The algorithm monitors the distances between rows and columns to ensure that the shortest paths are used for assignments. After this phase, all assignments are updated to reflect the minimum cost for each row in the cost matrix (Jonker & Volgenant, 1988, Jones et al., 2017, Fankhauser, S. et al., 2012). In summary, by integrating efficient initialization steps with augmenting path techniques, the LAPJV algorithm consistently achieves optimal assignments while significantly reducing computational time, especially in large-scale problems.

Algorithm 2 shows all the steps of finding a maximum matching.

3.2. **The proposed UFS2DPCA method algorithm.** The details of the steps of the proposed UFS2DPCA algorithm are shown in Algorithm 3. Furthermore, we solved an example to better demonstrate the proposed UFS2DPCA algorithm, and its details are shown in Figure 3. We Assume that $X \in \mathbb{R}^{5\times 3}$ is the input face image matrix.

In Figure 3 we have bolded the selected features.

4. Experiments

We discussed the efficiency of the proposed UFS2DPCA method in this section. So, we performed the experiments and then compared our UFS2DPCA

Algorithm 2 :**Maximum Weighted Bipartite Graph Matching using LAPJV Algorithm**

Input: n-by-n matrix C of weights $(w_{i,j})$ on the edges between sets Y and X. $X = \{x_1, x_2, x_3, \ldots, x_n\}$, $Y = \{y_1, y_2, y_3, \ldots, y_n\}$, $\forall i, j \in \{1, \ldots, n\} : w_{i,j} \geq 0$, n is the number of features, $|X| = |Y|$.
Output: indices: n indices of features that yield maximum total weight.

1) Begin
2) For $i \leftarrow 1$ **to** n **do**
Perform column reduction on matrix C
3) Reduction Transfer
Execute reduction transfer on matrix C
4) Augmenting Row Reduction
Perform augmenting row reduction on matrix C
5) Augmentation Phase
Initialize shortest path lengths and predecessor array for the shortest path tree
6) Find Minimum Shortest Path Lengths
Identify columns with new values for minimum shortest path lengths
7) Scan and Update
Scan a row of the matrix C
Update columns based on shortest path calculations
8) Determine Optimal Matching
Determine the optimal row and value for the maximum weight matching
End

algorithm with 8 popular unsupervised feature selection algorithms that consisted of UFSPCA-2022 (Beiranvand, Mehrdad, & Dowlatshahi, 2022), AEFS-2018 (Han, Wang, Zhang, Li, & Xu, 2018), DGUFS-2018 (Guo & Zhu, 2018), SCEFS-2021 (Xie, Wang, Xu, Huang, & Grant, 2021), SCAFS-2021 (Xie, Wang, Xu, Huang, & Grant, 2021), RSR-2015 (Zhu, Zuo, Zhang, Hu, & Shiu, 2015), SRCFS-2019 (Huang, Cai, & Wang, 2019), and URAFS 2018 (Li, Zhang, Zhang, Liu, & Nie, 2019) on four two-dimensional image datasets. In the following section, a description of the two-dimensional image datasets, the classifier, the evaluation criteria, and the results of the proposed UFS2DPCA method with other methods will be presented. It should be noted we implemented all algorithms by Matlab_R2018b.

4.1. **Datasets.** As mentioned, two-dimensional images are greyscale. That is, an image can be shown as $I \times I$ q-bit pixels. I denote the number of points along the axes and q manages the number of brightness values. In a greyscale image, q is 8, and the brightness level range is 2^q-1, which is shown in black and white with shades of gray in between, respectively. It should be noted that if I is too small, the lines will become visible as "blocks" and many details will

Algorithm 3 :**The proposed UFS2DPCA algorithm**

Input: Training two-dimensional image dataset,
Output: indexes of the selected features

1) Begin
2) Eigenvector Calculation:
Find the eigenvectors of the image dataset $P_{n\times n}$ using Algorithm 1.
3) Transform Data:
Transform P to a new space by computing $Y_{n\times m} = P_{n\times n}X'_{n\times m}$.
4) Calculate Pairwise Similarity:
Calculate the pairwise similarity between the original features of the images and the extracted features using the Pearson Correlation Coefficient (PCC).
5) Construct Weighted Bipartite Graph:
Construct a weighted bipartite graph $G = (V, E)$, $V = X \cup Y$, where E is the similarity matrix C.
6) Perform Graph Matching:
Execute $[c, T] = \text{LAPJV}(C)$ // calculated by Algorithm 2
7) Select Features:
The variable c contains the indexes of the selected features.
8) End

be lost. Large I values provide more detail, but due to a large number of pixels, more storage space is required and image processing takes longer (Nixon, 2007). We used four famous two-dimensional image datasets and described them in the following. Also, we present detailed characteristics of each two-dimensional image dataset in Table number one. The Yale dataset consists of 165 face images, which we cropped to 50 × 50 pixels (Kaggle, n.d., Tan et al., 2015). Similarly, the ORL dataset contains 400 images, cropped to 56 × 46 pixels (Kaggle, n.d., Tan et al., 2015). The Jaffe dataset includes 213 images, which were cropped to 26 × 26 pixels (Yuan et al., 2019). Lastly, the pixraw10P dataset comprises 100 face images, also cropped to 100 × 100 pixels (Jundong, n.d.).

4.2. **Classifier.** We use KNN to classify the test data. KNN works by using the distance function and measures the similarity between two samples (Jiang et al., 2007). K-Nearest Neighbors (KNN), at first, compute the distance between the new sample and the training sample and find K- Nearest Neighbors. Then, specify the class of the new sample relative to the class to which the neighbor belongs, if they all belong to the same class, the new sample also belongs to this class. Otherwise, each class is scored after selection and the new sample class is chosen in proportion to a specified formula (Wang, 2019). In this paper, we considered $k = 5$.

Table 1. Characteristics of the datasets.

Dataset	# Samples	# Features	# Classes	Domain
Jaffe	213	26×26	7	Image
Yale	165	50×50	15	Image
ORL	400	56×46	40	Image
pixraw10P	100	100×100	10	Image

4.3. **Evaluation.** We used NMI, F-measure, Recall, Precision, and Accuracy metrics to measure the effectiveness of our UFS2DPCA method with the 8 algorithms previously above explained. In the following, we describe each of these measurement criteria.

a) **Input image**: a two-dimensional gray level image of size 5*5, the image is from the Yale dataset.

$$\begin{bmatrix} 106.2513 & 163.7609 & 135.8620 & 122.5821 & 58.6359 \\ 117.0656 & 122.6800 & 136.3424 & 107.2944 & 72.0225 \\ 169.5375 & 175.4163 & 154.5865 & 166.1170 & 131.9058 \\ 162.1373 & 112.1959 & 92.3323 & 112.3185 & 164.7369 \\ 197.7514 & 147.6319 & 151.5969 & 134.5611 & 207.0926 \end{bmatrix}$$

b) Two-dimensional image matrix (P), double

$$\begin{bmatrix} 116.4605 & -246.8638 & 4.5306 & -7.1467 & -21.2983 \\ 132.4938 & -212.6798 & 26.1625 & 10.8584 & -21.2983 \\ 216.2736 & -284.6950 & -0.4462 & 11.2036 & -21.2983 \\ 234.8507 & -177.5049 & -4.5311 & 4.2886 & -21.2983 \\ 291.4776 & -242.6229 & 24.3260 & -2.4497 & -21.2983 \end{bmatrix}$$

e) $Y{=}P{*}X$ (new feature matrix from linear transform)

$$\begin{bmatrix} 0.4471 & 0.9071 & 0.7544 & 1.4129 & 1.4522 \\ 0.5137 & 0.9818 & 0.8239 & 1.3676 & 1.4063 \\ 0.4861 & 0.9288 & 0.8618 & 1.4715 & 1.4756 \\ 0.5803 & 1.0033 & 1.0168 & 1.5241 & 1.5074 \\ 0.5681 & 1.0210 & 0.9629 & 1.4882 & 1.5058 \end{bmatrix}$$

f) C matrix (correlation similarity)

[5 2 1 4 3]

g) Selected index of (P matrix) by LAPJV algorithm

$$\begin{bmatrix} 106.2513 & 163.7609 & \mathbf{135.8620} & 122.5821 & 58.6359 \\ 117.0656 & \mathbf{122.6800} & 136.3424 & 107.2944 & 72.0225 \\ 169.5375 & 175.4163 & 154.5865 & 166.1170 & \mathbf{131.9058} \\ 162.1373 & 112.1959 & 92.3323 & \mathbf{112.3185} & 164.7369 \\ \mathbf{197.7514} & 147.6319 & 151.5969 & 134.5611 & 207.0926 \end{bmatrix}$$

h) **Output** (selected features of matrix P)

FIGURE 3. We show the UFS2DPCA method by example, bold fonts are selected features.

The accuracy criterion is commonly used to measure classification performance, and its definition is given in the equation 10 (Tharwat, 2018):

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{10}$$

TP means it was a positive sample and we have classified it correctly. FN means it was a positive sample, but we classified it as negative. TN means that the sample was negative, and we classified it correctly. Lastly, FP indicates the sample was negative, but we mistakenly classified it as positive (Tharwat, 2018). Another widely used criterion is precision, and its definition is given in the equation (11) (Tharwat, 2018):

$$\text{Precision} = \frac{TP}{TP + FP} \tag{11}$$

The definition of the Recall measure is given in the equation (12) (Tharwat, 2018):

$$\text{Recall} = \frac{TP}{TP + FN} \tag{12}$$

F-measure is also an important evaluation criterion and is given in the equation (13) (Tharwat, 2018):

$$F\text{-measure} = \frac{TP}{\frac{1}{2}(FP + FN) + TP} \tag{13}$$

Normalized mutual information (NMI) (Beiranvand, Mehrdad, & Dowlatshahi, 2022) is the last measure that we used to evaluate the efficiency of our UFS2DPCA method. NMI evaluates the mutual information (MI in short) between predicted class labels and actual class labels. NMI is obtained from entropy (Beiranvand, Mehrdad, & Dowlatshahi, 2022). The entropy formula for variable B is given in the equation (14):

$$H(B) = -E\left[\log p(b)\right] = -\sum_{x} p(b) \log p(b) \tag{14}$$

The MI of two variables B and A is calculated by equation (15) (Beiranvand, Mehrdad, & Dowlatshahi, 2022):

$$I(B; A) = H(B) - H(B|A) \tag{15}$$

H (B) and H (A) indicate the entropies of B and A. NMI is calculated using equation (16):

$$\text{NMI}(B, A) = \frac{2I(B; A)}{H(B) + H(A)} \tag{16}$$

The value of NMI is between 0 and 1. A value of zero for NMI indicates that B and A are completely inconsistent, while a value of 1 indicates that B and A are identical (Beiranvand, Mehrdad, & Dowlatshahi, 2022).

We provide a concise overview of comparative methods utilized in unsupervised feature selection, along with their abbreviations and relevant references, as shown in Table 2. By presenting this information, readers can easily compare and understand the context of the proposed method within the broader landscape of existing techniques.

In all the evaluation tables and figures, all algorithms were run for 30 iterations for each image dataset. We use the average F-measure, average recall, average precision, average accuracy, and average NMI metrics to measure the quality of our UFS2DPCA method and other algorithms. We randomly split samples of each dataset into training and testing data, allocating 70% of the samples of each dataset to the training data and 30% to the test data.

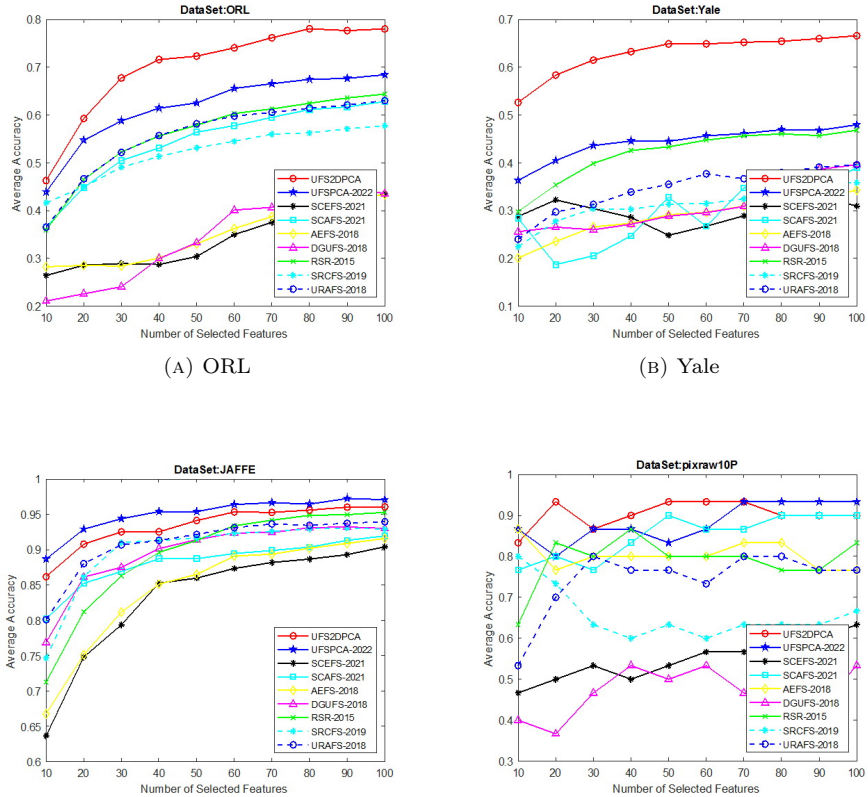
Table 2. Overview of Comparative Methods.

Abbreviation	**Full Name**	**Reference**
UFS2DPCA	Unsupervised Feature Selection using 2D PCA	[the proposed method]
UFSPCA	Unsupervised Feature Selection using Principal Component Analysis	Beiranvand et al.(2022)
AEFS	AutoEncoder Feature Selector	Han et al. (2018)
DGUFS	Dependence Guided Unsupervised Feature Selection	Guo et al. (2018)
SCEFS	Feature Selection via Standard deviation and Cosine similarity	Xie et al. (2021)
SCAFS	Standard deviation and Anti-Cosine similarity-based Feature Selection	Xie et al. (2021)
RSR	Regularized Self-Representation	Zhu et al. (2015)
SRCFS	Unsupervised Feature Selection Approach Based on Multi-Subspace Randomization and Collaboration	Huang et al. (2019)
URAFS	Uncorrelated Regression with Adaptive Graph Feature Selection	Li et al. (2018)

We select the top q features via each algorithm as the optimal feature selection subset, where $q \in \{10 : 10 : 100\}$, and execute classification on these subsets. We applied the proposed UFS2DPCA algorithm to four two-dimensional image datasets and then compared it with eight state-of-the-art unsupervised feature selection algorithms that were previously introduced. In all figures, we varied the number of selected features from 10 to 100 in steps of 10, illustrated on the horizontal axis. The results of average accuracy for all datasets and algorithms mentioned are shown in Figure 4. The vertical axis of all datasets illustrates the average accuracy. The average accuracy results for the UFS2DPCA algorithm on the ORL dataset show a clear increasing trend as the number of selected features increases from 10 to 100. This improvement indicates that the algorithm effectively enhances classification performance with

more features, reaching a peak accuracy of approximately 78.03% when selecting 100 features, suggesting a strong ability to leverage relevant information for improved outcomes. The average accuracy results for the UFS2DPCA algorithm on the Yale dataset indicate a gradual improvement as the number of selected features increases from 10 to 100. Achieving a maximum accuracy of approximately 66.60% at 100 features suggests that the algorithm is effective in enhancing classification performance. The average accuracy results for the UFS2DPCA algorithm on the Jaffe dataset show a consistent improvement as the number of selected features increases from 10 to 100. Reaching a maximum accuracy of approximately 96.08% at 100 features indicates that the algorithm effectively enhances classification performance, demonstrating its capability to leverage relevant features for optimal results. For the pixraw10P dataset, which is a high-dimensional dataset where each image has 100×100 pixels (resulting in 10,000 features), the accuracy values for the UFS2DPCA algorithm consistently improve as the number of selected features increases. Starting from 0.8333 at 10 features, the accuracy peaks at 0.9333 between 30 to 70 features. This stability in accuracy after 30 features suggests that the algorithm effectively selects relevant features, maintaining high and consistent performance across multiple independent runs. The upward trend of average accuracy figures across the four datasets indicates that the selected features are relevant.

The results of average precision for all face datasets and algorithms mentioned are shown in Figure 5. The vertical axis of all datasets illustrates the average precision. The average precision results for the UFS2DPCA algorithm on the ORL dataset demonstrate a consistent increase as the number of selected features rises from 10 to 100. This trend indicates that the algorithm successfully improves precision, achieving a maximum value of approximately 80.90% when utilizing 100 features, reflecting its effectiveness in enhancing the quality of classification outcomes. The average precision results for the UFS2DPCA algorithm on the Yale dataset also demonstrate a steady increase as the number of selected features rises from 10 to 100, with a maximum precision of approximately 70.59% at 100 features. These findings indicate that the algorithm effectively identifies relevant instances. On the Jaffe dataset, the average precision results show a consistent increase as the number of selected features rises from 10 to 100. Achieving a maximum precision of approximately 96.43% at 100 features reflects the algorithm's effectiveness in correctly identifying relevant instances, highlighting its strong performance in classification tasks. For the pixraw10P dataset, which is a high-dimensional dataset, the precision values for the UFS2DPCA algorithm exhibit a strong upward trend, starting from 0.7214 with 10 features and steadily increasing to 0.9667 at 40 features, where it remains consistent through 100 features. This indicates that the algorithm efficiently improves its ability to correctly identify relevant features with increasing feature subsets, achieving high precision and maintaining it across independent iterations. It is also clear from the figures that our proposed algorithm has a high precision value. As shown in Figure 5, the proposed method



(A) ORL (B) Yale

(C) Jaffe (D) pixraw10P

FIGURE 4. Average Accuracy

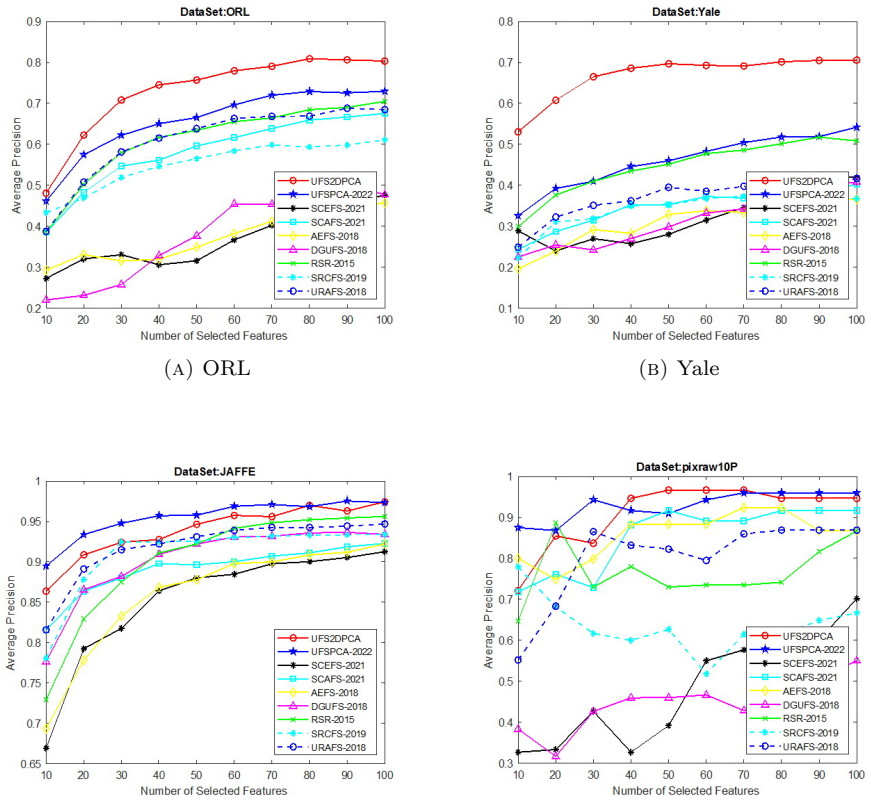
has performed well. The upward trend of average precision figures across the four datasets indicates that the selected features are relevant.

The results of the average Recall for all face datasets and algorithms mentioned are shown in Figure 6. The vertical axis of all datasets illustrates the average Recall. The average recall results for the UFS2DPCA algorithm on the Yale dataset indicate a gradual improvement as the number of selected features increases from 10 to 100. Achieving a maximum recall of approximately 69.16% at 100 features reflects the algorithm's effectiveness in identifying true positive instances. The average recall results for the UFS2DPCA algorithm on the ORL dataset show a significant upward trend as the number of selected features increases from 10 to 100. Achieving a maximum recall of approximately 82.06% with 100 features indicates the algorithm's effectiveness in accurately

identifying positive instances, highlighting its robust performance in classification tasks. The average recall results for the UFS2DPCA algorithm on the Jaffe dataset show a consistent upward trend as the number of selected features increases from 10 to 100. Reaching a maximum recall of approximately 96.48% at 100 features indicates that the algorithm effectively identifies true positive instances, demonstrating its robustness and efficiency in classification tasks. For the pixraw10P dataset, which is a high-dimensional dataset, the UFS2DPCA algorithm ranks among the best algorithms in several feature subsets. The recall values for the UFS2DPCA algorithm starts at 0.7250 with 10 features and increases significantly to 0.9600 at 40 features, maintaining this high level up to 100 features. This suggests that the algorithm effectively captures a large proportion of relevant features as the number of selected features grows, showing strong recall performance across independent runs. It is also clear from the figures that our proposed algorithm has a high Recall value. As shown in Figure 6, the UFS2DPCA method has performed well. The upward trend of average Recall figures across the four datasets indicates that the selected features are relevant.

The results of the average F-measure for all face datasets and algorithms mentioned are shown in Figure 7. The vertical axis of all datasets illustrates the average F-measure. The average F-measure results for the UFS2DPCA algorithm on the Yale dataset demonstrate a steady increase as the number of selected features rises from 10 to 100. With a maximum F-measure of approximately 69.77% at 100 features, these findings indicate a balanced performance in terms of precision and recall, suggesting that the algorithm is effective in achieving satisfactory classification outcomes. The average F-measure results for the UFS2DPCA algorithm on the ORL dataset indicate a Steady improvement as the number of selected features increases from 10 to 100. Reaching a peak F-measure of approximately 81.57% with 100 features demonstrates the algorithm's balanced.

performance in both precision and recall, highlighting its effectiveness in achieving a robust classification outcome. The average F-measure results for the UFS2DPCA algorithm on the Jaffe dataset show a consistent improvement as the number of selected features increases from 10 to 100. With a maximum F-measure of approximately 96.45% at 100 features, these findings highlight the algorithm's balanced performance in terms of precision and recall, indicating its effectiveness in achieving robust classification outcomes. In the pixraw10P dataset, the F-measure values for the UFS2DPCA algorithm demonstrate strong performance, starting at 0.7232 with 10 features and peaking at 0.9633 with 40 features, maintaining this high level through 100 features. This shows that the algorithm effectively balances precision and recall, achieving optimal feature selection after 40 features, with consistent performance across multiple independent runs. As shown in Figure 7, the UFS2DPCA method has performed well. The upward trend of average F-measure figures across the four



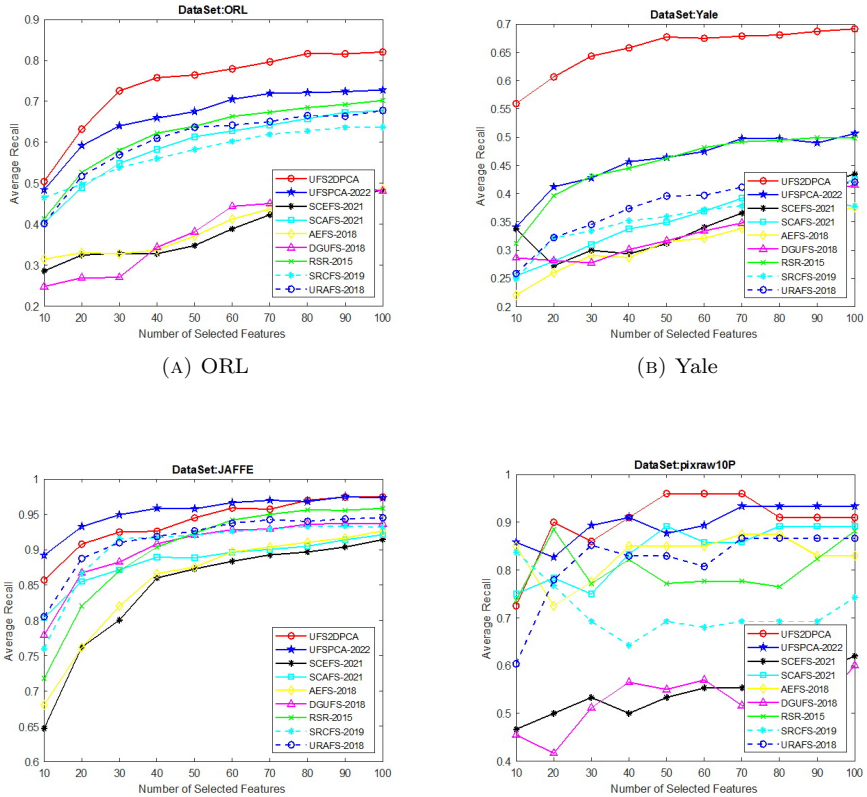
(A) ORL (B) Yale

(C) Jaffe (D) pixraw10P

FIGURE 5. Average Precision

datasets indicates that the proposed UFS2DPCA algorithm selects features that are entirely relevant.

The results of the average NMI for all face datasets and algorithms mentioned are shown in Figure 8. The vertical axis of all datasets illustrates the average NMI. The average NMI results for the UFS2DPCA algorithm on the Yale dataset show a consistent increase as the number of selected features rises from 10 to 100. Reaching a maximum NMI of approximately 76.28% at 100 features indicates that the algorithm effectively captures the underlying structure of the data, which enhances the classification performance and demonstrates its robustness in feature selection. The average NMI results for the UFS2DPCA algorithm on the ORL dataset demonstrate a significant increase as the number of selected features rises from 10 to 100. Achieving a maximum NMI of approximately 90.01% at 100 features indicates that the unsupervised feature



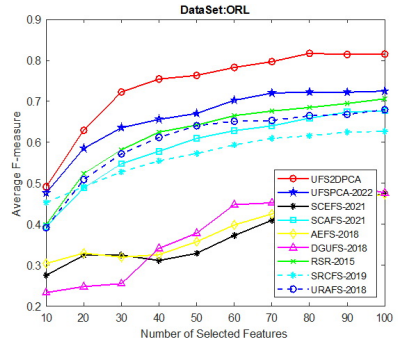
(A) ORL (B) Yale

(C) Jaffe (D) pixraw10P

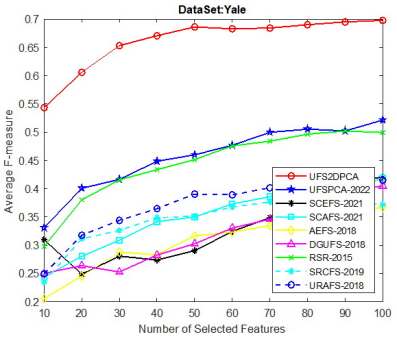
FIGURE 6. Average Recall

selection method Effectively enhances classification performance, showcasing the algorithm's ability to capture the underlying structure of the data without relying on class labels.

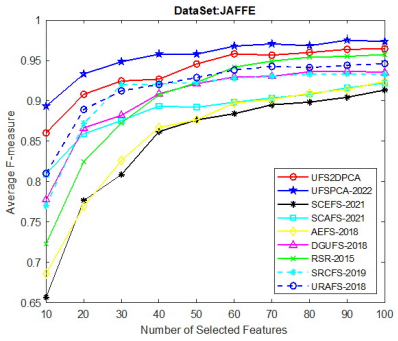
The average NMI results for the UFS2DPCA algorithm on the Jaffe dataset exhibit a clear increasing trend as the number of selected features rises from 10 to 100. Achieving a maximum NMI of approximately 95.75% at 100 features indicates that the proposed UFS2DPCA algorithm effectively captures the underlying structure of the data, enhancing the classification performance. In the pixraw10P dataset, which is a high-dimensional dataset, the average NMI across the selected features range from 10 to 100 is generally high and stable, reflecting the algorithm's effectiveness in retaining useful information. Additionally, the lack of significant fluctuation in the NMI values across different runs demonstrates the repeatability and robustness of the proposed method.



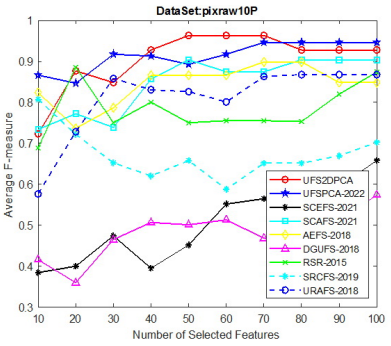
(A) ORL



(B) Yale



(C) Jaffe

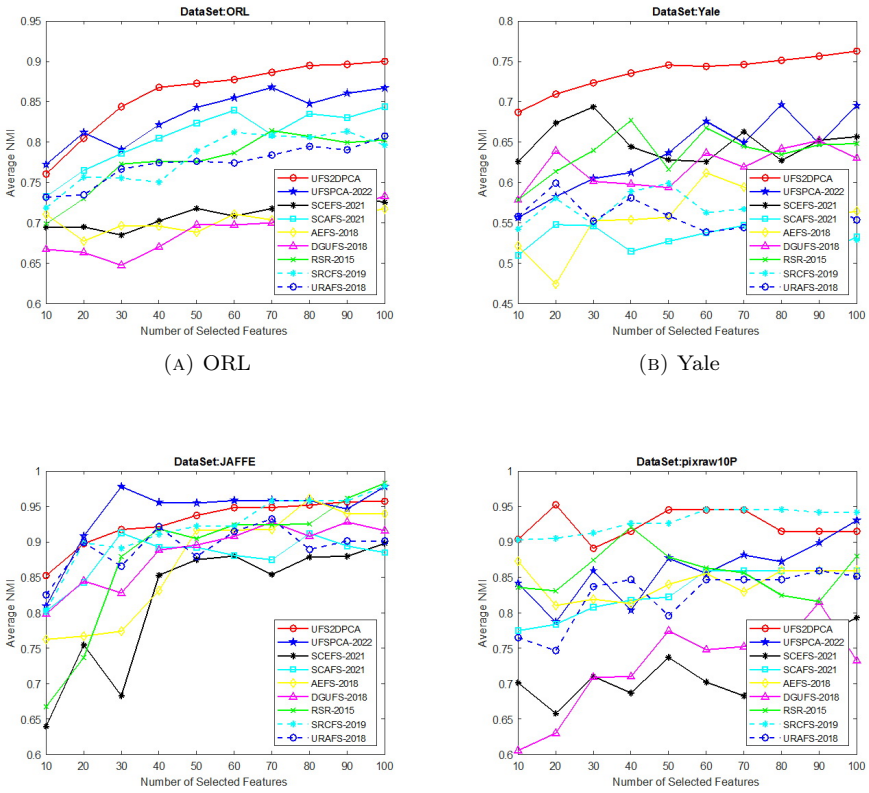


(D) pixraw10P

FIGURE 7. Average F-measure

It is also clear from the figures that our proposed algorithm has a high NMI value.

Tables 3 -7 report the average results of the 30 runs for all methods mentioned and our proposed method on each dataset, including the average F-measure, average recall, average precision, average accuracy, and NMI metrics over 10 final feature subsets for test data. The maximum results are highlighted in bold fonts. The results are analyzed using non-parametric statistical tests, specifically the Wilcoxon signed-rank (Beiranvand, Mehrdad, & Dowlatshahi, 2022) and the Friedman tests (Derrac et al., 2011). The Wilcoxon test compares two matched samples, while the Friedman test ranks multiple algorithms. In this study, a p-value of 0.05 is used, following the procedures outlined by Derrac et al. (2011).



(A) ORL (B) Yale

(C) Jaffe (D) pixraw10P

FIGURE 8. Average NMI

These results are validated using the non-parametric Friedman test (Derrac et al., 2011), which is the most widely recognized method for evaluating differences among more than two related samples. The Friedman test will be employed along with post-hoc procedures as a suitable complement to the Friedman-related analyses. Procedure for Conducting the Friedman Test:
The initial step in determining the Friedman test statistic involves transforming the original data into ranks. This process is accomplished through the following steps:

1. Collect the observed outcomes for each pair of algorithms and problems.

2. For each problem i, assign ranks to the values, starting from 1 (indicating the best outcome) to k (indicating the worst outcome). These ranks are represented as r_i^j (where $1 \leq j \leq k$).

3. For every algorithm j, compute the average of the ranks derived from all problems to establish the final rank as follows:

$$R_j = \frac{1}{n} \sum_i r_i^j \tag{17}$$

This methodology ranks the algorithms individually for each problem, ensuring that the algorithm with the highest performance receives a rank of 1, the next best gets a rank of 2, and so forth. In cases of ties, it is advisable to calculate the average ranks. Under the null hypothesis—which posits that all algorithms exhibit similar performance (implying their ranks R_j should be equivalent)—the Friedman statistic F_f can be computed accordingly:

$$F_f = \frac{12n}{k(k+1)} \left[\sum_j R_j^2 - \frac{k(k+1)^2}{4} \right] \tag{18}$$

where n represents the number of observations (or rows) in the dataset, k indicates the number of groups (or treatments), and R_j denotes the ranks for each group. The computed statistic F_f is approximately distributed according to a chi-squared statistic is denoted as χ^2 (chi-squared) distribution with k - 1 degrees of freedom, provided that n (the number of observations) and k (the number of groups) are sufficiently large. As a general rule, it is usually considered adequate for n>10 and k>5 (Derrac et al., 2011).

Table 3. Average Accuracy.

Dataset	UFS2DPCA	UFSPCA	SCEFS	SCAFS	AEFS	DGUFS	RSR	SRCFS
Jaffe26*26	0.9346	**0.9506**	0.8331	0.8829	0.8459	0.8963	0.8924	0.8987
Yale50*50	**0.6288**	0.4308	0.3273	0.3253	0.2941	0.3170	0.4295	0.3271
ORL56*46	**0.7013**	0.6171	0.3401	0.5444	0.3501	0.3420	0.5602	0.5218
pixraw10P	**0.9033**	0.8833	0.55	0.85	0.8033	0.4667	0.79	0.6567
Wilcoxon	+	+	+	+	+	+	+	+

Table 3 presents the average accuracy of various feature selection algorithms across multiple datasets. A detailed analysis of the performance is as follows: UFS2DPCA consistently outperforms other algorithms across all datasets. Notably, it achieves the highest accuracy on the Jaffe 26*26 and pixraw10P datasets, with scores of 0.9346 and 0.9033, respectively. These results indicate that UFS2DPCA is highly effective in selecting relevant features, even in high-dimensional settings, and maintains robust performance throughout. UFSPCA also performs exceptionally well on the Jaffe 26*26 dataset, achieving an accuracy of 0.9506, the highest among all the algorithms. However, it underperforms on other datasets, particularly on Yale 50*50, where its

accuracy drops to 0.4308, indicating a potential limitation in handling datasets with different structures or higher complexity. The SCEFS and SCAFS algorithms exhibit satisfactory performance on simpler datasets like Jaffe 26*26, with accuracies of 0.8331 and 0.8829, respectively. However, their performance deteriorates significantly on more complex datasets such as Yale 50*50, with accuracies dropping to 0.3273 and 0.3253, respectively. This suggests that these algorithms may not be well-suited for datasets with more intricate patterns or noise. AEFS shows moderate accuracy on certain datasets such as Jaffe 26*26 (0.8459) and pixraw10P (0.8033). However, its performance on Yale 50*50 and ORL 56*46 (0.2941 and 0.3501, respectively) indicates its struggles in handling more challenging or high-dimensional data. DGUFS consistently exhibits lower performance across all datasets. For example, its accuracy scores of 0.3170 on Yale 50*50 and 0.4667 on pixraw10P demonstrate its ineffectiveness in extracting relevant features from more complex data. RSR performs moderately well on simpler datasets such as Jaffe 26*26 (0.8924) and ORL 56*46 (0.5602). However, its performance on Yale50*50 remains suboptimal (0.4295), further indicating its limitations in handling higher-dimensional and more complex datasets. The SRCFS and URAFS algorithms achieve reasonable accuracy on Jaffe26*26 (0.8987 and 0.9103, respectively), but their performance deteriorates on more complex datasets such as Yale 50*50, with accuracy scores of 0.3271 and 0.3523, respectively. This suggests that while these algorithms may perform well on simpler datasets, they struggle to generalize to more intricate data. Among the compared algorithms, UFS2DPCA and UFSPCA demonstrate superior performance across the datasets, with UFS2DPCA providing the most consistent results. On the other hand, SCEFS, SCAFS, and AEFS show limited generalizability, especially when applied to more complex datasets. DGUFS and URAFS underperform in most scenarios, particularly in high-dimensional settings. This analysis highlights the importance of selecting an appropriate feature selection method based on the dataset's characteristics and the specific problem domain.

Furthermore, the results of the Friedman test and post-hoc analysis demonstrate the superior performance of the UFS2DPCA algorithm compared to other methods. According to the Friedman test, UFS2DPCA achieved an average accuracy of 0.9033 on the pixraw10P 100*100 dataset, significantly outperforming other feature selection methods. The p-value of 5.8874e-06 indicates statistically significant differences among the evaluated methods, clearly showing that UFS2DPCA outperforms algorithms such as UFSPCA, SCEFS, SCAFS, and others. The post-hoc analysis further confirms these findings, revealing significant differences between UFS2DPCA and UFSPCA with a p-value of 0.0000, as well as between UFS2DPCA and SCEFS with a p-value of 0.0056. These differences are attributable to UFS2DPCA's optimized feature selection capabilities and enhanced data processing techniques. In conclusion, the results of both the Friedman test and post-hoc analysis confirm that UFS2DPCA is the most effective method among the evaluated algorithms. The

results of the Wilcoxon statistical test also demonstrate the superiority of our proposed method.

Table 4. Average Precision.

Dataset	UFS2DPCA	UFSPCA	SCEFS	SCAFS	AEFS	DGUFS	RSR	SRCFS	URAFS
Jaffe	**0.9371**	**0.9548**	0.8524	0.8913	0.8591	0.9024	0.9019	0.9097	0.9191
Yale	**0.6683**	0.4599	0.3218	0.3447	0.3086	0.3134	0.4462	0.3415	0.3663
ORL	**0.7302**	0.6574	0.3670	0.5829	0.3723	0.3763	0.6117	0.5518	0.6104
pixraw10P	**0.9100**	**0.9295**	0.4827	0.8540	0.8581	0.4426	0.7668	0.6367	0.8017
Wilcoxon	+	+	+	+	+	+	+	+	+

Table 4 presents the average precision results of various unsupervised feature selection methods across four datasets: Jaffe, Yale, ORL, and pixraw10P. The results reveal notable differences in the effectiveness of these methods.
Jaffe Dataset: The UFS2DPCA and UFSPCA methods achieved the highest average precision, with values of 0.9371 and 0.9548, respectively. In contrast, other methods such as SCEFS and SCAFS exhibited lower performance, with average precision scores around 0.85.
Yale Dataset: For the Yale dataset, UFS2DPCA maintained the highest precision at 0.6683, though this represents a significant decrease compared to the Jaffe dataset. UFSPCA (0.4599) and other methods, including SCEFS (0.3218) and SCAFS (0.3447), displayed considerably lower precision levels.
ORL Dataset: Similar to the previous datasets, UFS2DPCA again recorded the best precision at 0.7302. Other methods, particularly SCEFS (0.3670) and SCAFS (0.5829), demonstrated consistently lower performance.
pixraw10P Dataset: In the pixraw10P dataset, UFS2DPCA (0.9100) and UFSPCA (0.9295) continued to show superior performance.Conversely, methods like SCEFS (0.4827) and DGUFS (0.4426) yielded lower precision values. The analysis reveals that UFS2DPCA consistently outperforms other methods across all datasets, indicating its effectiveness in selecting relevant features. The variability in precision across different datasets suggests that the performance of feature selection methods can be heavily influenced by the characteristics of the data. These findings underscore the necessity for further investigation into optimizing feature selection algorithms tailored to specific dataset types. Overall, the results provide a compelling case for the adoption of UFS2DPCA as a reliable method for enhancing feature selection in machine learning applications. Furthermore, the Friedman test revealed significant differences in the performance of the evaluated feature selection methods.
The results are as follows: Chi-Square Statistic = 27, Degrees of Freedom $df = 3$, and the p-value is 5.8874×10^{-6}. The low p-value indicates that at least one of the methods performs significantly differently from the others, specifically in terms of precision. The Friedman test and post-hoc analyses

further demonstrate notable differences among the feature selection methods. Although UFS2DPCA performs well. These results highlight UFS2DPCA as a robust and reliable method for feature selection. The results of the Wilcoxon statistical test also demonstrate the superiority of our proposed method.

Table 5. Average Recall.

Dataset	UFS2DPCA	UFSPCA	SCEFS	SCAFS	AEFS	DGUFS	RSR	SRCFS	URAFS
Jaffe	**0.9367**	**0.9544**	0.8434	0.8845	0.8558	0.9025	0.8999	0.9032	0.9158
Yale	**0.6557**	0.4566	0.3463	0.3516	0.3121	0.3328	0.4512	0.3509	0.3738
ORL	**0.7413**	0.6647	0.3819	0.5916	0.3951	0.3846	0.6199	0.5767	0.6035
pixraw10P	**0.9005**	0.8992	0.5433	0.84	0.8310	0.5195	0.8010	0.7132	0.8169
Wilcoxon	+	+	+	+	+	+	+	+	+

Table 5 presents the average recall results of various unsupervised feature selection methods across four datasets: Jaffe, Yale, ORL, and pixraw10P. Recall, as a measure of a method's ability to identify relevant features, is critical for assessing the effectiveness of feature selection techniques.

Jaffe Dataset: The UFS2DPCA and UFSPCA methods achieved the highest average recall, with values of 0.9367 and 0.9544, respectively. Other methods, such as SCEFS and SCAFS, showed lower performance, with recall scores around 0.84, indicating that the top-performing methods excel in retaining relevant features.

Yale Dataset: In the Yale dataset, UFS2DPCA recorded the highest average recall at 0.6557, though with a significant drop compared to the Jaffe dataset. The UFSPCA method (0.4566) and other methods, including SCEFS (0.3463) and SCAFS (0.3516), exhibited notably lower recall values, reflecting the challenges associated with feature selection in datasets with less distinct features.

ORL Dataset: The UFS2DPCA method maintained a strong average recall of 0.7413, outperforming other methods such as SCEFS (0.3819) and SCAFS (0.5916). This indicates that UFS2DPCA is robust in identifying relevant features across varied datasets. In the pixraw10P dataset, UFS2DPCA (0.9005) and UFSPCA (0.8992) continued to demonstrate superior performance in terms of recall. Conversely, methods such as SCEFS (0.5433) and DGUFS (0.5195) exhibited lower recall values, reinforcing the effectiveness of the top two methods. The analysis of average recall results demonstrates that UFS2DPCA consistently outperforms other feature selection methods across all datasets, underscoring its effectiveness in retaining relevant features. The variability in recall across different datasets suggests that the characteristics of the data significantly impact the performance of the feature selection methods. These findings highlight the importance of selecting appropriate feature selection algorithms based on the specific dataset characteristics. Overall, the results provide compelling evidence for the utilization of UFS2DPCA in enhancing

recall in machine learning applications, thereby improving the ability to effectively identify relevant features. The performance of various feature selection methods was assessed using recall metrics, and a Friedman test was applied to evaluate the differences among the methods. The results of the Friedman test are as follows: Chi-Square Statistic: 27, Degrees of Freedom (df): 3, p-value: 5.8874×10^{-6}. These results indicate a statistically significant difference in recall performance among the feature selection methods ($p < 0.001$), suggesting that at least one method performs differently from the others. Post-hoc comparisons were conducted to identify specific differences between the methods. The key findings include UFS2DPCA vs. UFSPCA: A significant difference was observed ($p < 0.001$), highlighting distinct recall performance between these two methods.UFS2DPCA vs. SCEFS: UFS2DPCA significantly outperformed SCEFS ($p = 0.0056$), indicating its superior ability to enhance recall. These results emphasize the effectiveness of UFS2DPCA as a robust feature selection method in terms of recall, particularly in outperforming SCEFS. The results of the Wilcoxon statistical test also demonstrate the superiority of our proposed method.

Table 6. Average F-measure.

Dataset	UFS2DPCA	UFSPCA	SCEFS	SCAFS	AEFS	DGUFS	RSR	SRCFS	URAFS
Jaffe	**0.9369**	**0.9546**	0.8476	0.8877	0.8572	0.9024	0.9008	0.9063	0.9173
Yale	**0.6609**	0.4562	0.3289	0.3483	0.3069	0.3186	0.4436	0.3442	0.3664
ORL	**0.7389**	0.6617	0.3706	0.5897	0.3837	0.3787	0.6200	0.5671	0.6043
pixraw10P	**0.9050**	**0.9140**	0.5063	0.8467	0.8441	0.4774	0.7833	0.6723	0.8088
Wilcoxon	+	+	+	+	+	+	+	+	+

Table 6 presents the average F-measure results for various unsupervised feature selection methods across four datasets: Jaffe, Yale, ORL, and pixraw10P. The F-measure is a critical metric that combines precision and recall, providing a comprehensive evaluation of the performance of the feature selection methods.

Jaffe Dataset: The UFS2DPCA and UFSPCA methods achieved the highest average F-measure scores of 0.9369 and 0.9546, respectively. In contrast, other methods such as SCEFS and SCAFS demonstrated lower performance, with F-measure scores around 0.85, indicating a consistent pattern of effectiveness among the top-performing methods.

Yale Dataset: In this dataset, UFS2DPCA again exhibited the highest average F-measure of 0.6609, although this reflects a notable decrease compared to the Jaffe dataset. The performance of UFSPCA (0.4562) and other methods, such as SCEFS (0.3289) and SCAFS (0.3483), remained significantly lower,

highlighting the challenges of feature selection in datasets with less distinguishable features. **ORL Dataset:** The UFS2DPCA method maintained a competitive average F-measure of 0.7389, outperforming other methods like SCEFS (0.3706) and SCAFS (0.5897). This indicates the robustness of UFS2DPCA across diverse scenarios.
pixraw10P Dataset: In the pixraw10P dataset, UFS2DPCA (0.9050) and UFSPCA (0.9140) continued to display superior performance. Conversely, methods such as SCEFS (0.5063) and DGUFS (0.4774) yielded lower F-measure values, further emphasizing the effectiveness of the top two methods. The results from the average F-measure analysis indicate that UFS2DPCA consistently outperforms other feature selection methods across all datasets, reaffirming its effectiveness in identifying relevant features. The variability in performance across different datasets suggests that the characteristics of the data significantly influence the effectiveness of the employed feature selection methods. These findings highlight the importance of selecting appropriate feature selection algorithms based on the specific characteristics of the dataset. Overall, the results provide strong support for the utilization of UFS2DPCA in enhancing the F-measure in machine learning applications, thereby improving the overall predictive performance of models. The Friedman test was conducted to assess the effectiveness of various feature selection methods based on F-measure across different datasets. The results revealed significant differences among the methods, indicated by a Chi-squared statistic of 27.00 and a corresponding p-value of 5.8874e-06. This strong evidence suggests that at least one of the methods performs differently from the others. The statistical significance of these findings underscores the potential of UFS2DPCA as an effective approach for enhancing feature selection in unsupervised learning scenarios. The results of the Wilcoxon statistical test also demonstrate the superiority of our proposed method.

Table 7. Average NMI.

Dataset	UFS2DPCA	UFSPCA	SCEFS	SCAFS	AEFS	DGUFS	RSR	SRCFS	URAFS
Jaffe	**0.9291**	0.9224	0.8197	0.8791	0.8728	0.8844	0.8827	0.9203	0.8932
Yale	**0.7362**	0.6359	0.6491	0.5300	0.5567	0.6191	0.6368	0.5617	0.5599
ORL	**0.8606**	0.8337	0.7102	0.8070	0.7031	0.6901	0.7765	0.7807	0.7736
pixraw10P	**0.9244**	0.8608	0.7195	0.8305	0.8421	0.7230	0.8582	0.9296	0.8244
Wilcoxon	+	+	+	+	+	+	+	+	+

Table 7 presents the average Normalized Mutual Information (NMI) results for various unsupervised feature selection methods across four datasets: Jaffe, Yale, ORL, and pixraw10P. NMI is a crucial metric for evaluating the amount of information shared between the selected features and the ground

truth, making it an essential measure for assessing the quality of feature selection methods.

Jaffe Dataset: The UFS2DPCA method achieved the highest average NMI of 0.9291, closely followed by UFSPCA at 0.9224. Other methods, such as SCEFS (0.8197) and SCAFS (0.8791), demonstrated significantly lower performance, indicating that the top-performing methods effectively preserve relevant information from the data.

Yale Dataset: In the Yale dataset, UFS2DPCA again exhibited the highest NMI at 0.7362. UFSPCA (0.6359) and other methods, including SCEFS (0.6491) and SCAFS (0.5300), displayed relatively lower NMI values, highlighting the challenges in capturing mutual information in datasets with less distinct features.

ORL Dataset: The UFS2DPCA method maintained a robust average NMI of 0.8606, outperforming other methods such as SCEFS (0.7102) and SCAFS (0.8070). This underscores the capability of UFS2DPCA to effectively capture mutual information across diverse datasets.

pixraw10P Dataset: In the pixraw10P dataset, UFS2DPCA (0.9244) and UFSPCA (0.8608) continued to demonstrate strong performance in terms of NMI. Other methods, such as SCEFS (0.7195) and DGUFS (0.7230), yielded lower values, further emphasizing the effectiveness of the leading methods.

The average NMI analysis indicates that UFS2DPCA consistently outperforms other feature selection methods across all datasets, demonstrating its effectiveness in retaining relevant information. The variability in NMI across different datasets suggests that the characteristics of the data significantly influence the performance of the feature selection methods utilized. These findings highlight the importance of selecting suitable feature selection algorithms based on the specific attributes of the dataset. Overall, the results provide compelling evidence for the adoption of UFS2DPCA in enhancing NMI in machine learning applications, thereby improving the overall quality of feature selection. The Friedman test statistics indicate a Chi-squared value of 25.9333, accompanied by a p-value of 9.8489e-06. This low p-value demonstrates a statistically significant difference among the algorithms being compared, suggesting that at least one of the feature selection methods outperforms the others. The results of the Wilcoxon statistical test also demonstrate the superiority of our proposed method.

4.4. **Computational Complexity.** Running time is denoted by 'O (f)' which f is a function of the input size (Manual, n.d.). The proposed UFS2DPCA method has three computation steps: step one is to calculate the feature matrix using 2DPCA. The complexity of 2DPCA is The complexity of 2DPCA is 2DPCA $= O(Nm^2n + m^3)$ (Tan et al., 2015). m and n are the dimensionality of the row and column of a two-dimensional matrix, respectively. N denotes the number of samples. In the second step, the Pearson correlation matrix is O (Nd^2) (Matching, 1995). d is the dimensionality of the sample vector

such that $d = mn$. In the last step, the total computational complexity of weighted bipartite graph matching is O ($|V|^3$) =O(d^3) (Beiranvand, Mehrdad, & Dowlatshahi, 2022). Finally, after simplification, the total complexity of the proposed UFS2DPCA algorithm becomes O(Nd^2)+ O(d^3).

5. Conclusion

We proposed a new matrix-based algorithm called UFS2DPCA. In the proposed algorithm, we first directly extract uncorrelated and orthogonal features from the face image dataset using 2DPCA. Then, we calculate the similarity between the extracted features and the original features. Finally, we model two sets of feature matrices and the similarity between them into weighted bipartite graph matching, providing a robust representation of the features. The selected features, obtaining the best features using the LAPJV algorithm, are a subset of the original features in maximum matching and do not exhibit any correlation. Our proposed method can select features with high accuracy, leveraging the hidden knowledge of features from feature extraction methods. Additionally, modeling the problem into a weighted bipartite graph offers an attractive representation of the features. A series of experimental results have shown that UFS2DPCA outperforms other compared algorithms. While we designed this method for unsupervised feature selection, it can be adapted to a supervised feature selection method by replacing the 2DPCA method with the 2DLDA method, and so forth. In future work, we intend to generalize the proposed method for selecting models of nonlinear structures such as 2DLLE. Furthermore, our proposed method can be adapted to a semi-supervised feature selection framework. It can also be transformed into a one-dimensional and two-dimensional supervised feature selection algorithm by substituting its feature extraction method with LDA or 2DLDA, respectively. In upcoming papers, we will discuss the use of evolutionary algorithms, such as the ant colony algorithm, as an alternative to the LAPJV algorithm.

6. Aknowledgement

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

7. Author Contributions:

Conceptualization, Vahid Mehrdad and Mohammad Bagher Dowlatshahi; methodology, Firoozeh Beiranvand, Mohammad Bagher Dowlatshahi; software, Firoozeh Beiranvand; data curation, Firoozeh Beiranvand; writing—original draft preparation, Firoozeh Beiranvand; writing—review and editing, Firoozeh Beiranvand and Mohammad Bagher Dowlatshahi; supervision, Vahid Mehrdad and Mohammad Bagher Dowlatshahi. All authors have read and agreed to the published version of the manuscript.

8. 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.

9. Ethical considerations

The authors avoided data fabrication and falsification.

10. Funding

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

11. Conflict of interest

The authors declare no conflict of interest

References

[1] Ahamad, M., & Ahmad, N. (2021). Machine Learning Approaches to Digital Learning Performance Analysis. *International Journal of Computing and Digital Systems, 10*(1), 963–971. https://doi.org/10.12785/IJCDS/100187

[2] 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, 109085.* https://doi.org/10.1016/j.knosys.2022.109085

[3] Derigs, U. (1981). A shortest augmenting path method for solving minimal perfect matching problems. *Networks, 11*(4), 379–390. https://doi.org/10.1002/net.3230110407

[4] Derrac, J., García, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation, 1*(1), 3–18. https://doi.org/10.1016/j.swevo.2011.02.002

[5] Fankhauser, S., Riesen, K., Bunke, H., & Dickinson, P. (2012). Suboptimal graph isomorphism using bipartite matching. *International Journal of Pattern Recognition and Artificial Intelligence, 26*(6), 1–26. https://doi.org/10.1142/S0218001412500139

[6] Guo, J., & Zhu, W. (2018). Dependence guided unsupervised feature selection. 32nd AAAI Conference on Artificial Intelligence, 2232–2239.

[7] Guyon, I. (2008). Practical feature selection: From correlation to causality. PASCAL Network. http://eprints.pascal-network.org/archive/00004038

[8] 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, 2941–2945. https://doi.org/10.1109/ICASSP.2018.8462261

[9] Hashemi, A., Bagher, M., & Nezamabadi-pour, H. (2020). MFS-MCDM: Multi-label feature selection using multi-criteria decision making. *Knowledge-Based Systems, 206*, 106365. https://doi.org/10.1016/j.knosys.2020.106365

[10] Hashemi, A., Bagher, M., & Nezamabadi-pour, H. (2021a). VMFS: A VIKOR-based multi-target feature selection. *Expert Systems with Applications, 182*, 115224. https://doi.org/10.1016/j.eswa.2021.115224

[11] Hashemi, A., Bagher, M., & Nezamabadi-pour, H. (2021b). 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
[12] Hashemi, A., Dowlatshahi, M. B., & Nezamabadi-Pour, H. (2021). A bipartite matching-based feature selection for multi-label learning. *International Journal of Machine Learning and Cybernetics, 12*(2), 459–475. https://doi.org/10.1007/s13042-020-01180-w
[13] Hashemi, A., Dowlatshahi, M. B., & Nezamabadi-pour, H. (2021). Ensemble of feature selection algorithms: A multi-criteria decision-making approach. *International Journal of Machine Learning and Cybernetics*. https://doi.org/10.1007/s13042-021-01347-z
[14] Huang, D., Cai, X., & Wang, C. D. (2019). Unsupervised feature selection with multi-subspace randomization and collaboration. *Knowledge-Based Systems, 182*. https://doi.org/10.1016/j.knosys.2019.07.027
[15] Huang, P., & Yang, X. (2022). Unsupervised feature selection via adaptive graph and dependency score. *Pattern Recognition, 127*, 108622. https://doi.org/10.1016/j.patcog.2022.108622
[16] https://www.kaggle.com/datasets/tavarez/the-orl-database-for-training-and-testing
[17] https://jundongl.github.io/scikit-feature/datasets.html
[18] Jiang, L., Cai, Z., Wang, D., & Jiang, S. (2007). Survey of improving K-nearest-neighbor for classification. Proceedings - Fourth International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2007, 1, 679–683. https://doi.org/10.1109/FSKD.2007.552
[19] Jones, W., Chawdhary, A., & King, A. (2017). Optimising the Volgenant–Jonker algorithm for approximating graph edit distance. *Pattern Recognition Letters, 87*, 47–54. https://doi.org/10.1016/j.patrec.2016.07.024
[20] Jonker, R., & Volgenant, T. (1988). A shortest augmenting path algorithm for dense and sparse linear assignment problems. In DGOR/NSOR: Papers of the 16th Annual Meeting of DGOR in Cooperation with NSOR/Vorträge der 16. Jahrestagung der DGOR zusammen mit der NSOR 1988 (pp. 622-622). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-73778-7-164
[21] Karami, S., Saberi-Movahed, F., Tiwari, P., & Marttinen, P. (2023). Unsupervised feature selection based on variance–covariance subspace distance. *Neural Networks, 166*, 188–203. https://doi.org/10.1016/j.neunet.2023.06.018
[22] Khalid, S., Khalil, T., & Nasreen, S. (2014). A survey of feature selection and feature extraction techniques in machine learning. In Proceedings of 2014 Science and Information Conference (SAI 2014) (pp. 372–378). https://doi.org/10.1109/SAI.2014.6918213
[23] Kong, H., Wang, L., Teoh, E. K., Li, X., Wang, J. G., & Venkateswarlu, R. (2005). Generalized 2D principal component analysis for face image representation and recognition. *Neural Networks, 18*(5–6), 585–594. https://doi.org/10.1016/j.neunet.2005.06.041
[24] Li, J., Chen, J., Qi, F., Dan, T., Weng, W., Zhang, B., Yuan, H., Cai, H., & Zhong, C. (2022). Two-dimensional unsupervised feature selection via sparse feature filter. *IEEE Transactions on Cybernetics*, 1–13. https://doi.org/10.1109/TCYB.2022.3162908
[25] Li, J., Liang, X., & Li, P. (2020). Two-dimensional semi-supervised feature selection. *Proceedings of the Conference*, 280–287.
[26] 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
[27] Lu, H., Plataniotis, K. N., & Venetsanopoulos, A. N. (2011). A survey of multilinear subspace learning for tensor data. *Pattern Recognition, 44*(7), 1540–1551.

https://doi.org/10.1016/j.patcog.2011.01.004
[28] Hashemi, A., Dowlatshahi, M. B., & Nezamabadi-Pour, H. (2021). A bipartite matching-based feature selection for multi-label learning. *International Journal of Machine Learning and Cybernetics*, 12(2), 459–475. https://doi.org/10.1007/s13042-020-01180-w
[29] Yang, J., Zhang, D., Frangi, A. F., & Yang, J. Y. (2004). Two-Dimensional PCA: A New Approach to Appearance-Based Face Representation and Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1), 131–137. https://doi.org/10.1109/TPAMI.2004.1261097
[30] Miao, J., & Niu, L. (2016). A survey on feature selection. *Procedia Computer Science, 91*, 919–926. https://doi.org/10.1016/j.procs.2016.07.111
[31] Nixon, M. (2007). Feature extraction & image processing (2nd ed.).
[32] Paniri, M., Dowlatshahi, M. B., & Nezamabadi-pour, H. (2021). Ant-TD: Ant colony optimization plus temporal difference reinforcement learning for multi-label feature selection. *Swarm and Evolutionary Computation, 64*, 100892. https://doi.org/10.1016/j.swevo.2021.100892
[33] Rahimi, Z., Taghipour, K., Khadivi, S., & Afhami, N. (2012). Document and sentence alignment in comparable corpora using bipartite graph matching. 2012 6th International Symposium on Telecommunications, IST 2012, 817–821. https://doi.org/10.1109/ISTEL.2012.6483098
[34] Sanguansat, P. (Ed.). (2012). Principal component analysis. BoD–Books on Demand.
[35] Schrijver, A. (2012). A course in combinatorial optimization.
[36] Tabakhi, S., Moradi, P., & Akhlaghian, F. (2014). An unsupervised feature selection algorithm based on ant colony optimization. *Engineering Applications of Artificial Intelligence, 32*, 112–123. https://doi.org/10.1016/j.engappai.2014.03.007
[37] Tan, H., Zhang, X., Guan, N., Tao, D., & Huang, X. (2015). Two-Dimensional Euler PCA, 548–559.
[38] Tharwat, A. (2018). Classification assessment methods. *Applied Computing and Informatics, 17*(1), 168–192. https://doi.org/10.1016/j.aci.2018.08.003
[39] Vasudev, C. (2006). Graph theory with applications. New Age International.
[40] Wahid, A., Khan, D. M., Hussain, I., Khan, S. A., & Khan, Z. (2022). Unsupervised feature selection with robust data reconstruction (UFS-RDR) and outlier detection. *Expert Systems with Applications, 201*, 117008. https://doi.org/10.1016/j.eswa.2022.117008
[41] Wang, C., Wang, J., Gu, Z., Wei, J. M., & Liu, J. (2024). Unsupervised feature selection by learning exponential weights. *Pattern Recognition, 148*, 110183. https://doi.org/10.1016/j.patcog.2023.110183
[42] Wang, C., Yin, B., Bai, X., & Sun, Y. (2008). Color face recognition based on 2DPCA. *Proceedings - International Conference on Pattern Recognition*, 8–11. https://doi.org/10.1109/ICPR.2008.4761093
[43] Wang, L. (2019). Research and Implementation of Machine Learning Classifier Based on KNN. *IOP Conference Series: Materials Science and Engineering, 677*(5), 0–5. https://doi.org/10.1088/1757-899X/677/5/052038
[44] Wang, Q., & Gao, Q. (2016). Robust 2DPCA and its application. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 1152–1158. https://doi.org/10.1109/CVPRW.2016.147
[45] Wang, X., Li, H., Zhou, Y., & Zheng, H. (2021). Lanczos algorithm for 2DPCA. *Journal of Physics: Conference Series, 2025*(1). https://doi.org/10.1088/1742-6596/2025/1/012035
[46] 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, 112379. https://doi.org/10.1016/j.knosys.2024.112379
[47] Xie, J., Wang, M., Xu, S., Huang, Z., & Grant, P. W. (2021). The unsupervised feature selection algorithms based on standard deviation and cosine similarity for genomic data analysis. *Frontiers in Genetics, 12*, 684100. https://doi.org/10.3389/fgene.2021.684100
[48] Yang, J., & Liu, C. (2007). Horizontal and vertical 2DPCA-based discriminant analysis for face verification on a large-scale database. *IEEE Transactions on Information Forensics and Security, 2*(4), 781–792. https://doi.org/10.1109/TIFS.2007.910239
[49] Yang, J., Zhang, D., Frangi, A. F., & Yang, J. Y. (2004). Two-dimensional PCA: A new approach to appearance-based face representation and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 26*(1), 131–137. https://doi.org/10.1109/TPAMI.2004.1261097
[50] Yu, L., & Liu, H. (2004). Efficient feature selection via analysis of relevance and redundancy. *The Journal of Machine Learning Research, 5*, 1205–1224.
[51] Yuan, H., Li, J., Lai, L. L., & Tang, Y. Y. (2019). Joint sparse matrix regression and nonnegative spectral analysis for two-dimensional unsupervised feature selection. *Pattern Recognition, 89*, 119–133. https://doi.org/10.1016/j.patcog.2019.01.014
[52] Yuan, H., Li, J., Lai, L. L., & Tang, Y. Y. (2020). Low-rank matrix regression for image feature extraction and feature selection. *Information Sciences, 522*, 214–226. https://doi.org/10.1016/j.ins.2020.02.070
[53] Zhang, F., Yang, J., Qian, J., & Xu, Y. (2015). Nuclear norm-based 2-DPCA for extracting features from images. *IEEE Transactions on Neural Networks and Learning Systems, 26*(10), 2247–2260.
[54] Zhou, X., & Wang, J. (2015). Feature selection for image classification based on a new ranking criterion. *Journal of Computer and Communications, 3*(3), 74–79. https://doi.org/10.4236/jcc.2015.33013
[55] 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

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

Vahid Mehrdad
Orcid number: 0000-0003-2121-9119
Department of Electrical Engineering
Lorestan University
Khoramabad, Iran
Email address: mehrdad.v@lu.ac.ir

Mohammad Bagher Dowlatshahi
Orcid number: 0000-0002-4862-0626
Department of Computer Engineering
Lorestan University
Khoramabad, Iran
Email address: dowlatshahi.mb@lu.ac.ir