High-Resolution Remote Sensing Image Classification with Kernel Linear Discriminant Analysis

K. Vinodha and E. S. Gopi

Contents

Abstract

Remote sensing image classification is difficult due to the curse of extracting valuable information from high-resolution satellite images in numerous applications such as land cover mapping, urban planning, and environmental monitoring. This study proposes a novel approach for high-resolution remote sensing image classification by utilizing kernel linear discriminant analysis (K-LDA). This chapter also provides a comprehensive theoretical background on principal component analysis (PCA) and linear discriminant analysis (LDA) to establish the foundation for our proposed method. Principal component analysis (PCA) is a widely used dimensionality reduction technique that aims to transform the

K. Vinodha (✉) · E. S. Gopi
Electronics and Communication Engineering, National Institute of Technology Tiruchirappalli, Tiruchirappalli, Tamil Nadu, India
e-mail: esgopi@nitt.edu

A. J. Kulkarni, A. H. Gandomi (eds.), *Handbook of Formal Optimization*,
https://doi.org/10.1007/978-981-19-8851-6_9-1

original data into a new space of uncorrelated variables, known as principal components. This dimensionality reduction simplifies the data and retains the most significant variance within the dataset. However, PCA does not consider class information, which can be crucial for classification tasks. On the other hand, linear discriminant analysis (LDA) considers class information. It aims to find a projection that maximizes the separation between different classes while minimizing the variance within each class. LDA has shown excellent performance in various classification tasks. Still, its linear nature can limit its effectiveness in handling complex and nonlinear data distributions in high-resolution remote sensing images. It has been introduced that kernel linear discriminant analysis (K-LDA) extends LDA into a high-dimensional feature space using kernel functions to address this limitation. K-LDA can capture intricate nonlinear relationships within the data by employing kernels, making it particularly well suited for classifying high-resolution remote sensing images. The proposed K-LDA approach is evaluated using a benchmark dataset of high-resolution remote sensing images(landcover satellite Images). Our experimental results demonstrate that K-LDA consistently outperforms the classification of different classes, especially when dealing with complex spatial patterns and diverse land covers. The proposed technique underscores the effectiveness of K-LDA for high-resolution remote sensing image classification tasks. Furthermore, the study provides insights into each technique's computational efficiency and resource demands, which are crucial for real-world applications where accuracy and efficiency are paramount. The implications of the findings extend to urban planning, environmental monitoring, and disaster response, where accurate land cover classification serves as a cornerstone. The step-by-step methodology described in the chapter serves as the foundation for tackling high-dimensional, multi-class data challenges in image classification tasks. This work advances the field's understanding of the intersection between image classification and remote sensing, fostering the development of more accurate and efficient land cover analysis methodologies.

Keywords

PCA · LDA · KLDA · HRRS images

Introduction

Image classification is an essential task in computer vision and pattern recognition. In data processing, hyperspectral remote sensing images (HRSIs) have a relatively small number of labelled samples and high spectral dimensionality, resulting in the Hughes phenomenon. Therefore, feature extraction is necessary before terrain classification (Liu et al. 2022; Li et al. 2022). However, dealing with high-dimensional image data can be challenging due to its complexity and computational requirements (Liu et al. 2022; Xin-Yi et al. 2020; Hidalgo et al. 2015).

Dimensionality reduction techniques, such as PCA, LDA, and KLDA, can help address these challenges by reducing the dimensionality (Jan et al. 2023; Gongjin et al. 2022; Otani et al. 2020; Santhi et al. 2022; Varma et al. 2019; Raghavendra et al. 2012; Li et al. 2020; Jia et al. 2022; JayaBrindha and Subbu 2018) of the data while preserving its essential features.

Earlier Study

The PCA dimensionality reduction technique, according to the author (Tsoulfidis and Athanasiadis 2022), creates important features by linearly reducing correlated variables into fewer uncorrelated variables. In this study, PCA is used to determine the top two (or a maximum of three) eigenvalues that compress the majority of data influencing the movement of the economy. These eigenvalues determine the price-profit or wage-profit curves' shapes. The authors use the Silhouette method and the k-means clustering algorithm to identify industry groups. The Silhouette approach is utilized to create the industry clusters, and the k-means clustering technique aids in extracting the ideal number of groupings of industries. The PCA method is used in the study to group industries and rank them according to their economic impact. The Silhouette method and the k-means clustering approach are used to identify and create clusters of industries.

The authors (Liu and Chen 2005) of this research use principal component analysis with linear discriminant analysis (PCA+LDA) as one of the face recognition techniques. The small sample size (SSS) issue with conventional LDA is addressed by the two-stage feature extraction method PCA+LDA. The face samples are projected onto a principal component analysis (PCA) subspace in the first stage of PCA+LDA. An unsupervised dimensionality reduction method called PCA seeks to identify the data's most significant changes. The dimensionality of the data is decreased, and the most distinct features are kept by projecting the face samples onto the PCA subspace. The projected data are subjected to linear discriminant analysis (LDA) in the second stage. With LDA, supervised dimensionality reduction increases the between-class distribution while reducing the within-class scatter. LDA improves classification performance by maximizing the between-class distribution, which increases the separability between various classes. PCA+LDA combines PCA and LDA, a well-liked method for face recognition problems because it effectively achieves feature extraction and dimensionality reduction. To further improve the classification performance, the research authors suggest integrating PCA+LDA with resampling LDA/QR. In his paper, a PCA+LDA feature extraction method is used overall. PCA is used in the first stage to lower the dimensionality of the face samples, and LDA is used in the second stage to increase the discriminative strength of the features.

This chapter's study (Sadeghi et al. 2001) focuses on the use of principal component analysis (PCA), linear discriminant analysis (LDA), and support vector machines (SVM) for face verification. The feature extraction methods are PCA and LDA, while the classifier is SVM. A dimensionality reduction method called PCA isolates the most important changes in the data. The dimensionality is decreased, yet the most distinctive traits are kept when the face samples are projected onto a subspace. On the other hand, LDA is a supervised dimensionality reduction strategy that improves the separability between various classes by maximizing the between-class scatter and minimizing the within-class scatter. The efficacy of face verification is enhanced by the effective feature extraction and dimensionality reduction made possible by the PCA and LDA combination used in this work. PCA and LDA improve the discriminative strength of the features while reducing the dimensionality, making face recognition more precise and dependable. The SVM classifier uses the retrieved features to distinguish between real and fake faces. SVM is renowned for its resilience against overfitting and capacity to handle high-dimensional data. The authors want to increase the precision and generalizability of the face verification system by using SVM as the classifier. The complementing roles played by PCA, LDA, and SVM in face verification are significant. PCA and LDA extract discriminative features and decrease the dimensionality of the data, providing an effective and precise representation of face photos. As a robust classifier, SVM successfully distinguishes between real and fake faces using the retrieved characteristics, producing accurate results for face verification. Using a mix of principal component analysis (PCA), linear discriminant analysis (LDA), and support vector machine (SVM), the research done in this chapter (Raghavendra et al. 2012) involves the development of a revolutionary face recognition system. Dimension reduction using PCA, feature extraction using LDA, and SVM classification comprise the approach's three components. SVM is used for classification because it reduces misclassification brought on by classes that are not linearly separable. Face picture classification based on computed LDA features is performed using the SVM algorithm. Finally, this method combines the advantages of PCA, LDA, and SVM to improve face recognition, especially in conditions with few samples per class and high-dimensional face images.

Using a transfer learning method, the work in the paper (Xin-Yi et al. 2020) involved classifying the land cover on high-resolution remote sensing (HRRS) photographs. Pseudo-label assignment and joint fine-tuning procedures were combined in the study's methodology. The researchers put out a semi-supervised method that connected the inherent properties of ground objects by using the high-level deep features of multi-source data. This method enabled automatically classifying unlabeled target photos without needing target domain-specific annotation data. The effectiveness of the suggested technique was tested in experiments using datasets of various classifications and sizes. The researchers examined and addressed the impact of several parameters, such as patch size, segmentation technique, and thresholds of the transfer learning scheme. The classification outcomes were contrasted with those of other methods, including multilayer perceptrons (MLP), support vector machines (SVM), random forests (RF), and maximum likelihood classification (MLC). The effectiveness of the various techniques was evaluated using metrics like overall accuracy and Kappa. The studies' outcomes showed how well and broadly the suggested method for classifying land cover in various HRRS photos worked. The fine-tuning strategies enhanced classification performance by acquiring structural knowledge and spatial linkages particular to the target area. Particularly when the spectral responses of the target and source domains were similar, the method demonstrated promising accuracy. To automatically categorize land cover in high-resolution remote sensing photos, the study included joint fine-tuning approaches, pseudo-label assignment, and a transfer learning strategy. The suggested method outperformed existing classification techniques, demonstrating the promise of this approach for real-time land-cover classification applications.

The earlier study provides a comprehensive understanding of applying deep learning techniques to high-dimension image data (satellite imagery for land cover analysis) (Rajeswara Rao et al. 2022; Ara et al. 2023; Xin-Yi et al. 2020; Raghavendra et al. 2012; Sadeghi et al. 2001; Liu and Chen 2005; Tsoulfidis and Athanasiadis 2022; Guo et al. 2019; Hasanlou et al. 2015). It covers a range of deep learning algorithms commonly used for image classification. The number of spectral bands or channels in the data cube is called the hyperspectral data's dimensionality. High-dimensional data are produced via hyperspectral data, often including hundreds of spectral bands. This high dimensionality might be problematic for classification systems because it can result in overfitting and the dimensionality curse.

Foundational Techniques for Data Analysis

Principal Component Analysis (PCA)

Overview of PCA

PCA is widely used as a dimensionality reduction technique in statistics and multivariate analysis. It was first introduced by Karl Pearson in 1901 and later independently developed by Harold Hotelling in the 1930s (Tsoulfidis and Athanasiadis 2022; Shemul et al. 2022). The PCA technique reduces the dimensionality of data by constructing relevant features by linearly combining the original data. In order to build relevant features, correlated variables are linearly transformed into fewer uncorrelated variables. The transformation is achieved by projecting the data into the reduced PCA space using the covariance or correlation matrix's eigenvectors, also called the principal components (PCs). Based on the projected data, most, if not all, of the variance can be captured by linear combinations of the initial data. It identifies the directions along which the data varies the most and provides a compact representation that retains the essential information. PCA can effectively reduce the computational complexity and improve the efficiency of subsequent analysis. This transformation is accomplished by finding the eigenvectors of the covariance matrix, which stand in for the principal components. Data compression, signal processing, and image processing are a few of the areas where PCA has found use. To classify high-dimensional image data more effectively and accurately, PCA is essential.

The main goal of PCA is to identify the directions in the initial feature space along which data differs most. Principal components are a set of directions that are orthogonal to one another and ranked according to how much variance in the data they explain.

In the context of dimensionality reduction, consider two random vectors, **u** and **v**, with m elements each, where n < m. The objective is to obtain a transformation matrix **W** of size m × m that maps the random vector **u** to **v** as $\mathbf{v} = \mathbf{W}^T\mathbf{u}$. This process is commonly known as the dimensionality reduction technique.

Steps Involved and an Analysis of Reduced Dimensions Using PCA

Let me consider the dataset represented as a matrix **X** of size $n \times m$, where n is the number of samples and m is the number of features. $\boldsymbol{\mu}$ is the mean vector of the dataset calculated by taking the column-wise mean of **X**. $\boldsymbol{\mu}$ is subtracted from each sample in **X** to center the data around the origin. Covariance matrix **C** of the centered data is computed by using the formula $\mathbf{C} = \frac{1}{n-1}(\mathbf{X} - \boldsymbol{\mu})^T(\mathbf{X} - \boldsymbol{\mu})$. Eigenvalue decomposition is performed on **C** to obtain the eigenvectors and eigenvalues.

The eigenvectors and eigenvalues are represented as **V** and $\boldsymbol{\Lambda}$, respectively. The eigenvalues are sorted in descending order. Arrange the corresponding eigenvectors accordingly. The decision on how many components or eigenvalues to fix in PCA is arbitrary. The amount of variance described by each component, the targeted dimensionality reduction, and the particular requirements of the problem are some variables that may affect dimensionality. The approach used to determine the optimal number of components is to plot the cumulative explained variance ratio as a function of the number of components and select the number of components that capture a significant portion of the variance. It helps balance dimensionality reduction and preserving important information. k eigenvectors corresponding to the k highest eigenvalues are selected to form the principal components matrix **P**. The centered data **X** is projected onto the principal components by computing $\mathbf{X}_{\text{reduced}} = (\mathbf{X} - \boldsymbol{\mu}) \cdot \mathbf{P}$. Finally, it returns the reduced-dimensional representation of the dataset $\mathbf{X}_{\text{reduced}}$.

The most significant information is kept, while the less important components are removed in the reduced-dimensional dataset produced by PCA. It can be

Algorithm 1 Principal Component Analysis (PCA)

Require: **train_data**: Training data matrix of size $n \times m$, where n and m is the number of samples and features respectively.
Ensure: **T**: Training data matrix after dimensionality reduction using PCA.
1: Center the training data by subtracting the mean of each feature.
2: Compute the covariance matrix of the centered training data.
3: Perform eigenvalue decomposition on the covariance matrix to obtain the eigenvectors **V** and eigenvalues **D**.
4: Sort the eigenvectors based on the corresponding eigenvalues in descending order.
5: Compute the cumulative sum of the eigenvalues and normalize it by the sum of all eigenvalues to obtain the explained variance ratio.
6: Determine the number of components to keep by finding the minimum number of components that capture a specified amount of variance (e.g., 95%).
7: Select the top (number of components) eigenvectors from **V** as the PC.
8: Compute the reduced training data **T** by projecting the centered training data onto the selected principal components.
9: **return** PM = (**V**, explained variance ratio) and **T**.

used as an input for other machine learning algorithms, feature extraction, or visualization. The PCA works best with linear relationships and may perform poorly with nonlinear ones. Techniques for nonlinear dimensionality reduction, such as the kernel technique, might be more suitable in such circumstances.
where PM → PCA model, T → reduced train data.

Illustration with example if a dataset is an image of size $n \times m$, where n and m are the number of samples and features, respectively. **u** and **v** are the number of elements in the vectors represented as n × m, representing the pixel values. Lowering the dimensionality while keeping the essential information for accurate image classification is advised because high-dimensional data is computationally demanding and may cause overfitting during classification tasks.

The aim is to find the transformation matrix **W** of size $n \times m$ to achieve dimensionality reduction. This transformation matrix will map the original vector **u** to the reduced vector **v**. The image is represented in a lower-dimensional space via the modified vector **v** while keeping the essential data for classification.

Now, to compute the variance and covariance: **v** is the variance of the transformed random variable, denoted as σ_v^2, which measures the spread of data in the reduced lower-dimensional space. It can be calculated as: $\sigma_v^2 = \mathbb{E}[\mathbf{v}^2] - (\mathbb{E}[\mathbf{v}])^2$.

The covariance between the **u** elements characterizes how they vary. The covariance matrix $\mathbf{C_u}$ of the random vector **u** is given by: $\mathbf{C_u} = \mathbb{E}[\mathbf{uu}^T] - \mathbb{E}[\mathbf{u}]\mathbb{E}[\mathbf{u}]^T$.

Here, $\mathbb{E}[\mathbf{v}^2]$ is the expected value (mean) of the element-wise square of **v**, and $\mathbb{E}[\mathbf{v}]$ is the expected value (mean) of **v**. Similarly, $\mathbb{E}[\mathbf{uu}^T]$ represents the expected value of the outer product of **u** with itself, and $\mathbb{E}[\mathbf{u}]$ is the mean vector of **u**.

By computing the variance and covariance, valuable insights into the spread and correlations of data points in the reduced lower-dimensional space are gained, aiding in understanding the impact of dimensionality reduction on the dataset.

Linear Discriminant Analysis

With $\mathrm{j} = 1, \ldots, \mathrm{K}$ and $\mathrm{i} = 1, \ldots, \mathrm{n_j}$, let $\mathbf{X_{ij}}$ represent the ith vector in the jth class. Calculate the between-class and within-class scatter matrices to analyze the data distribution. $\mathbf{S}_\mathrm{b}$ is the between-class scatter matrix and is computed as follows:

$$\mathbf{S_b} = \sum_{k=1}^{K} n_k (\mathbf{c_k} - \mathbf{c})^T (\mathbf{c_k} - \mathbf{c}) \tag{1}$$

where the centroid of the kth class is represented as $\mathbf{c_k}$ and the overall centroid **c** is computed as:

$$\mathbf{c} = \frac{1}{K} \sum_{i=1}^{K} \mathbf{c_i} \tag{2}$$

Similarly, the within-class scatter matrix, denoted as $\mathbf{S_W}$, is computed as follows:

$$\mathbf{S_W} = \sum_{j=1}^{K} \sum_{i=1}^{n_j} (\mathbf{x_{ij}} - \mathbf{c_j})^T (\mathbf{x_{ij}} - \mathbf{c_j}) \tag{3}$$

where $\mathbf{c_j} \rightarrow$ centroid of the jth class index.

To reduce the dimensionality of the data while preserving the class separability, $\mathbf{W} \rightarrow$ transformation matrix is introduced to map the original vectors to a lower-dimensional space: $\mathbf{y} = \mathbf{Wx}$. The scatter matrices are then transformed accordingly, resulting in $\mathbf{WS_bW^T}$ and $\mathbf{WS_wW^T}$.

To maximize the class separability, optimal $\mathbf{W}$ is obtained by finding the eigenvectors corresponding to the significant eigenvalues of the matrix $\mathbf{S_W^{-1}S_B}$. The matrix $\mathbf{W}$ (size r $\times$ m), where m and r are the dimensions before and after projection, is formed by row-wise arranging the appropriate eigenvectors. r is the number of significant eigenvalues corresponding to the eigenvectors.

The generated matrix $\mathbf{W}$ is the basis for linear discriminant analysis (LDA), ensuring that the projected vectors within each class are closer together and the centroids of various classes are distinguished.

LDA aims to find a linear projection that maximizes class separation while minimizing the variation within each class (Refer Algorithm 2).

$\boldsymbol{\mu}_c$ is the class means for each class c where, $\boldsymbol{\mu} = \frac{1}{N_c} \sum_{i=1}^{N_c} \mathbf{x}_i$. The overall mean $\boldsymbol{\mu}$ is computed for the training data as $\boldsymbol{\mu} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i$. $\mathbf{S}_b$ refers to between-class scatter matrix, where $\mathbf{S}_b = \sum_{c=1}^{C} N_c(\boldsymbol{\mu}_c - \boldsymbol{\mu})(\boldsymbol{\mu}_c - \boldsymbol{\mu})^T$ and the within-class scatter matrix is $\mathbf{S}_w = \sum_{c=1}^{C} \sum_{i=1}^{N_c} (\mathbf{x}_i - \boldsymbol{\mu}_c)(\mathbf{x}_i - \boldsymbol{\mu}_c)^T$. The eigenvectors and eigenvalues of $\mathbf{S}_w^{-1}\mathbf{S}_b$ are computed. The eigenvectors are sorted in descending order of their eigenvalues. The first d eigenvectors are selected corresponding to the d largest eigenvalues. Finally, the LDA projection matrix $\mathbf{W}$ is constructed using the selected eigenvectors.

Algorithm 2 Linear Discriminant Analysis (LDA)

Require: Training data $\mathbf{X}$, Training labels $\mathbf{Y}$
Ensure: LDA projection matrix $\mathbf{W}$

1: Compute the class means $\boldsymbol{\mu}_c$ for each class c
2: Compute the overall mean $\boldsymbol{\mu}$ of the training data
3: Calculate $\mathbf{S}_b$ the between-class scatter matrix.
4: Calculate $\mathbf{S}_w$ within-class scatter matrix.
5: Compute the matrix $\mathbf{S}_w^{-1}\mathbf{S}_b$
6: Calculate the eigenvalues and of $\mathbf{S}_w^{-1}\mathbf{S}_b$
7: Sort the eigenvectors in descending order of their eigenvalues
8: Select the first d eigenvectors corresponding to the d largest eigenvalues
9: Construct the LDA projection matrix $\mathbf{W}$ using the selected eigenvectors
10: **return W**

Kernel Linear Discriminant Analysis (KLDA)

Ronald A. Fisher created Linear Discriminant Analysis(LDA), a traditional linear feature extraction method, in 1936. It seeks to identify a lower-dimensional subspace that maximizes the data's ability to be divided into multiple classes. However, LDA assumes that the data can be separated linearly, which may not be true for complicated datasets. Kernel linear discriminant analysis combines kernel methods and linear discriminant analysis, which was presented as a solution to this problem. Before completing LDA, KLDA uses kernel functions to translate the data into a higher-dimensional feature space more likely to be linearly separable. Several pattern recognition tasks, including picture classification, have shown the effectiveness of KLDA. KLDA enhances class separability and classification accuracy by increasing the feature space's discriminative strength. KLDA focuses on maximizing the separability between different classes in the data by finding a low-dimensional representation that enhances the discriminative power (Liu et al. 2022; Xiaoming et al. 2012; Christopher 2006; Gopi 2020; Martinez and Kak 2001). It maps the data into a higher-dimensional feature space using a kernel function and performs LDA to extract discriminant features.

The Main Contribution of the Work

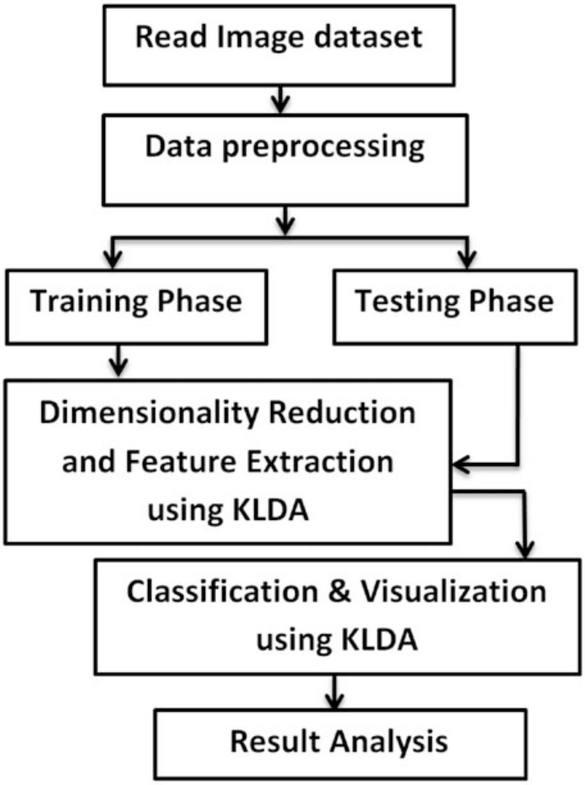
In this chapter, we leverage the strengths of KLDA for image classification. The proposed technique experiments with the landcover dataset (Xin-Yi et al. 2020; Xin-Yi Tong et al. 2020; Wang et al. 2008). The dataset has a multiclass of RGB images. PCA is applied for dimensionality reduction to capture the most important variations and performs KLDA for further feature extraction to enhance the separability between different image classes. The reduced-dimensional features are then used to train a binary SVM for classification. The classification accuracy is evaluated for each class, providing insights into the effectiveness of the PCA-KLDA approach for image classification tasks.

No attempts are made to use KLDA as a classification technique. In our proposed work, KLDA is used for dimensionality reduction, followed by feature extraction and a framework for image classification. The proposed method leverages the strengths of KLDA to address challenges in high-dimensional multi-class data classification. It offers a promising approach for improved image recognition and pattern recognition applications. It achieves a more efficient and discriminative representation of the image data, leading to improved classification performance. Furthermore, an experiment with real-time landcover dataset (Xin-Yi et al. 2020; Xin-Yi Tong et al. 2020) illustrates the application of KLDA image classification.

Methodology

This chapter proposes a way for utilizing KLDA to categorize multiclass data. The process for classifying high-dimensional multiclass data is shown in the block diagram (Refer Fig. 1). The block diagram (Refer Fig. 1) represents the pattern recognition task through a multistage process involving training and testing phases. The training phase begins with loading reference images and corresponding ground-truth labels. These images are partitioned into regions of interest. The subblock of size 10 × 10 with overlapping is taken from the reference image to train the network. A classification scheme is devised to assign labels to pixels in the images. Extracted image regions and their corresponding labels are collected. Dimensionality reduction is then performed using Eigen analysis of the collected data. During the testing phase, new images are processed similarly. Regions of interest are extracted, and the classification scheme is applied to predict labels for pixels. These predictions are compared with a predefined set of classes, and each region is assigned the most likely class. The results are visualized alongside the test images and their ground-truth labels. The KLDA method employs Gaussian kernels and Eigen analysis to create feature vectors for training and testing images,

Fig. 1 Block diagram



enabling effective pattern recognition. Kernel LDA is a nonlinear extension of linear discriminant analysis (LDA). It allows for better handling of complex, nonlinearly separable data by transforming it into a higher-dimensional feature space using kernel functions.

Training images are loaded, and regions of interest are extracted from them. The RGB values of pixels in these regions are normalized and treated as feature vectors. The feature vectors are transformed into a higher-dimensional space for each region using Gaussian kernel functions. A matrix "K" is formed where each column corresponds to a transformed feature vector. The "K" matrix is used to calculate class-specific means ($M1,\ M2,\ \ldots,\ M6$) and within-class scatter matrices (SW). A between-class scatter matrix (SB) is calculated using the class means. The generalized eigenvalue problem is solved to find the eigenvectors and eigenvalues of the matrix $((SW)^{-}1 \times SB)$). The top eigenvectors are retained for dimensionality reduction.

The eigenvectors obtained from the eigenvalue problem project the training data (matrix "K") to a lower-dimensional space. The projected data is stored in the matrix "V", where each column represents a transformed feature vector in the lower-dimensional space.

Testing images are loaded, and regions of interest are extracted. Similar to the training phase, Gaussian kernel functions are applied to transform the pixel values of these regions into the higher-dimensional space. The transformed data is projected into the lower-dimensional space using the same eigenvectors "E" obtained during training. Distances between the projected test data and the class-specific means ($V1,\ V2,\ \ldots,\ V6$) are calculated. For each region, the minimum distance indicates the class that best represents that region.

The resulting class predictions are visualized alongside the corresponding test images and ground-truth labels. The color map is used to represent different classes in the visualization.

Kernel Function

The kernel function used is a Gaussian kernel. The Gaussian kernel function computes the kernel value between two feature vectors.

$$\text{res} = \exp\left(-\frac{(x-y)^{\mathsf{T}}(x-y)}{\sigma}\right) \tag{4}$$

The hyperparameter sigma is tuned by varying the value from 0.001 to 10000, and it is found that at $\sigma = 100$, it achieves better classification results. The workflow describes kernel linear discriminant analysis for classifying images into multiple classes. It involves training with kernel transformations, dimensionality reduction, and subsequent testing and visualization of the classification results.

Kernel Linear Discriminant Analysis (KLDA)

The proposed technique employed is kernel-LDA to enhance the separability of vectors in the feature dimensional space, which maps the vectors to higher-dimensional space and then back to lower-dimensional space, respectively. Let **x** be transformed to the higher dimensional space as ψx, and the LDA is constructed as follows:

$$\mathbf{S_w^{\psi} w^{\psi}} = \mathbf{S_b^{\psi} w^{\psi}} \tag{5}$$

Here, $\mathbf{S_w^{\psi}}$ $\rightarrow$ within-class scatter matrix calculated using the vectors in the higher-dimensional space, and $\mathbf{S_b^{\psi}}$ $\rightarrow$ between-class scatter matrix in the higher-dimensional space. The LDA basis vector in this space is $\mathbf{w^{\psi}}$. To find $\mathbf{w^{\psi}}$, we represent it as $\mathbf{M^{\psi} u^{\psi}}$, where $\mathbf{M^{\psi}}$ is obtained from the matrix:

$$\mathbf{M} = [\psi(\mathrm{x}_{11}); \psi(\mathrm{x}_{12}); \psi(\mathrm{x}_{13}); \ldots; \psi(\mathrm{x}_{1\mathbf{n}_1}); \psi(\mathrm{x}_{21}); \psi(\mathrm{x}_{22})] \tag{6}$$

Next, we multiply ψ^{T} on both sides of the matrix equation, resulting in:

$$\psi^{\mathbf{T}} \mathbf{S}_{\mathbf{W}}^{\psi} \psi = \psi^{\mathbf{T}} \mathbf{S}_{\mathbf{B}}^{\psi} \psi \tag{7}$$

The matrix $\psi^{\mathbf{T}} \mathbf{S}_{\mathbf{W}}^{\psi} \psi$ corresponds to the between-class scatter matrix computed in the kernel space. This kernel space is spanned by the column space of matrix **G**, where the (m × n)th column vector of **G** is given by:

$$[\mathrm{k}(\mathrm{x}_{11}, \mathrm{x}_{\mathrm{mn}}) \ldots \mathrm{k}(\mathrm{x}_{\mathrm{nk}}\mathrm{K}, \mathrm{x}_{\mathrm{mn}})] \tag{8}$$

The kernel function is k(x, y). The eigenvector ψ is the kernel-LDA basis. The kernel-LDA basis is used to build the column vectors of the transformation matrix (**E**) (column-wise ordered eigenvectors (eigenvectors $\psi_1, \psi_2, \psi_3, \ldots, \psi_r$). In order to use kernel-LDA to map an arbitrary x with size m × n to the lower-dimensional vector, perform these steps:

- Map the vector **x** to the kernel space as $[\mathbf{k}(\mathbf{x}_{11}, \mathbf{x}); \mathbf{k}(\mathbf{x}_{11}, \mathbf{x}) \ldots \mathbf{k}(\mathbf{x}_{1_{\mathbf{n}1}}, \mathbf{x}) \ldots \mathbf{k}(\mathbf{x}_{\mathbf{n_k}}, \mathbf{x})]$. Let it be $\mathbf{k}(\mathbf{x})$.
- The obtained vector is then mapped to the lower-dimensional space using $\mathbf{E^T k(X)}$.
- Consequently, the vector in the feature dimensional space with size m × 1 is mapped to the lower-dimensional space with size r × 1.

Dataset Description and Preparation

The GID (Geographical Image Dataset) (Xin-Yi et al. 2020; Xin-Yi Tong et al. 2020) dataset, which contains 60 high-resolution GF-2 (Gaofen-2) images, was employed in the study. GID was created to represent the actual distribution of different types of land cover and to serve as training data for designed classifier models. The dataset covers a wide range of land-cover categories and geographic areas, allowing for comprehensive analysis and evaluation of land-cover classification algorithms. It provides a diverse set of images with different spatial resolutions and spectral bands, enabling researchers to study the impact of these factors on classification performance. GID has been made available online to benefit researchers in the field. It is a valuable resource for training and testing deep learning models for land cover classification in high-resolution remote sensing images.

Our research uses the land cover collection of HRRS images (Xin-Yi et al. 2020; Xin-Yi Tong et al. 2020). Recently, many high spatial-resolution remote sensing (HRRS) images have been made accessible for mapping land cover. However, finding an effective method for achieving accurate land-cover classification with high-resolution and heterogeneous remote sensing images is frequently challenging due to the complex information brought on by the increased spatial resolution and the data disturbances caused by various image acquisition conditions. The dataset comprises images of size 1000 $\times$1000 with five significant classes, such as built-up, Farmland, Forest, Meadow, and water, as class 1, 2, 3, 4, and 5, respectively. Additionally, the class 6 is considered as the class that does not belong to any of the five major classes (Refer Fig. 2). The dataset has reference images and corresponding ground-truth images.

From each class image (class 1 to 6), subblocks of size 9×9 are randomly selected. Each class has 600 subblock images, so the size of the input image is 3600, stored as input, and corresponding labels are stored in the output array.

Application of KLDA to the HRRS Images

KLDA is a feature extraction technique that aims to find a low-dimensional subspace that maximizes class separability. It combines the ideas of linear discriminant analysis (LDA) and the kernel trick to handle nonlinear data (Refer Algorithm 3). The distance matrix is computed to measure the pairwise distances between classes.

K is a kernel matrix. In KLDA, the kernel matrix captures the similarity between data points. Given a dataset $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$, where $x_i \in \mathbb{R}^d$, the kernel matrix **K** is defined as:

$$\mathbf{K}_{ij} = k(x_i, x_j) \tag{9}$$

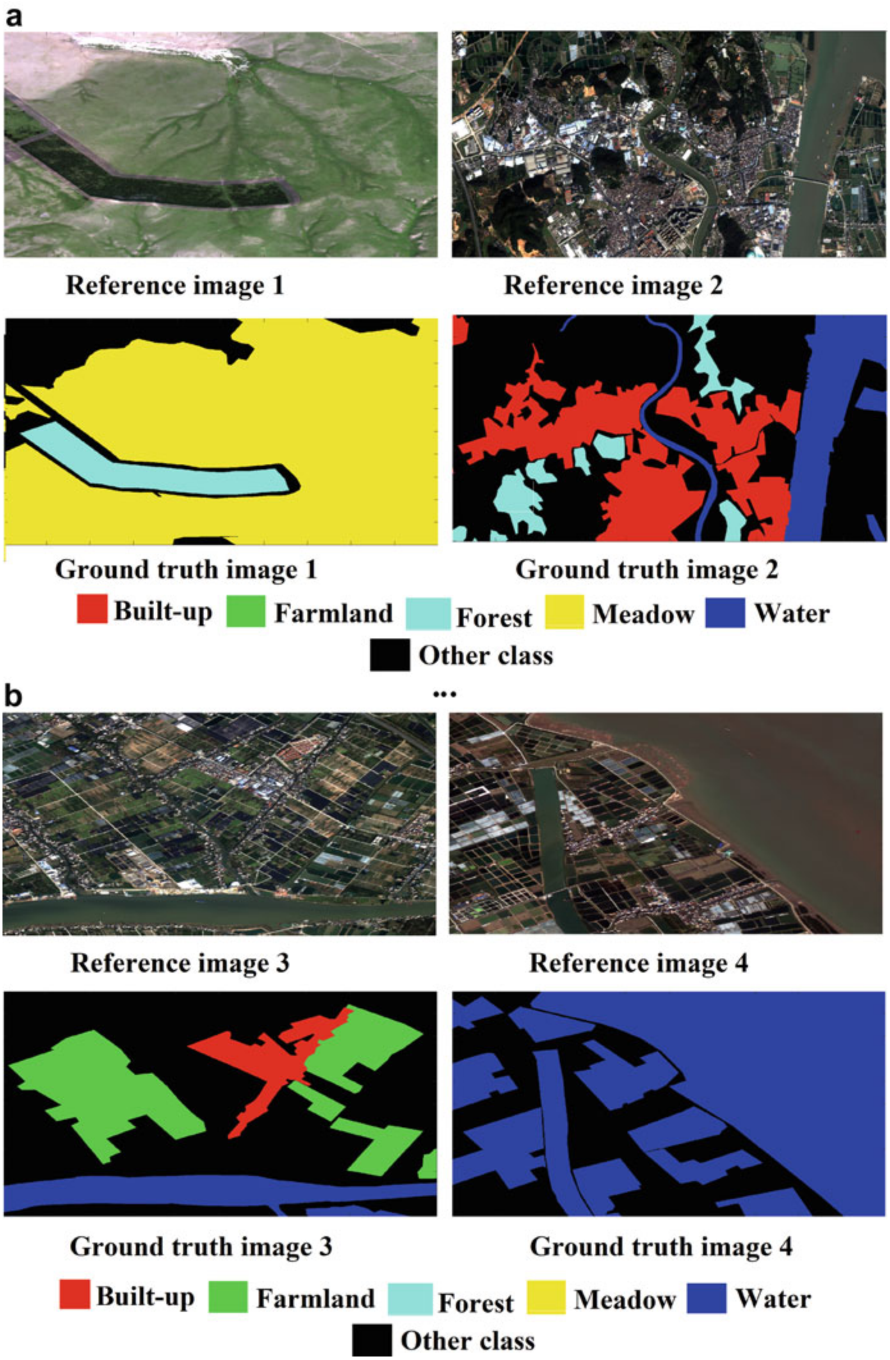


Fig. 2 Figure (**a**) and (**b**) represents sample Landcover dataset with reference images and corresponding ground truth images (Refer Xin-Yi Tong et al. 2020)

Algorithm 3 Kernel linear discriminant analysis (KLDA)

Require: Training data $\mathbf{X}$, Training labels $\mathbf{Y}$
Ensure: KLDA projection matrix $\mathbf{W}$

Compute the kernel matrix $\mathbf{K}$ using a kernel function
Compute the class means $\boldsymbol{\mu}_c$ for each class c in the kernel space
Compute the overall mean $\boldsymbol{\mu}$ of the training data in the kernel space
Compute the $\mathbf{S}_b$ in the kernel space
Compute the $\mathbf{S}_w$ in the kernel space
Compute the matrix $\mathbf{S}_w^{-1}\mathbf{S}_b$ in the kernel space
Calculate the eigenvectors and eigenvalues of $\mathbf{S}_w^{-1}\mathbf{S}_b$
Sort the eigenvectors in descending order of their eigenvalues
Select the first d eigenvectors corresponding to the d largest eigenvalues
Construct the KLDA projection matrix $\mathbf{W}$ using the selected eigenvectors
return W

The kernel function used is Gaussian (RBF) kernel, which computes the dot product between two vectors, equivalent to the inner product in the original feature space. It is a popular choice for linearly separable data or when the data exhibits a linear relationship.

$\boldsymbol{\mu}_c$ is the class means for each class c in the kernel space. $\boldsymbol{\mu}$ is the overall mean of the training data in the kernel space. $\mathbf{S}_b$ is the between-class scatter matrix in the kernel space. $\mathbf{S}_w$ is the within-class scatter matrix in the kernel space. $\mathbf{S}_w^{-1}\mathbf{S}_b$ is the matrix computed in the kernel space. The eigenvectors and eigenvalues are given as $\mathbf{S}_w^{-1}\mathbf{S}_b$. In KLDA, the number of eigenvectors to retain is determined by the number of classes minus one (n − 1). KLDA aims to find a projection that maximizes the between-class scatter while minimizing the within-class scatter. The resulting eigenvalues and eigenvectors correspond to the discriminant directions that best separate the classes. The eigenvectors chosen are 14 and are sorted in descending order of their eigenvalues. The first d eigenvectors are selected to correspond to the d largest eigenvalues. Finally, the KLDA projection matrix $\mathbf{W}$ is constructed using the selected eigenvectors.

The distance matrix is computed to measure the pairwise distances between classes in the dataset (refer to Fig. 3). It provides valuable information about the separability of different classes. The Euclidean distance is used, which calculates the straight-line distance between two points in an Euclidean space. Let two points in an n-dimensional space, as $(x_1, x_2, \ldots, x_n)$ and $(y_1, y_2, \ldots, y_n)$; the Euclidean distance (D) between them is given by:

$$D = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots + (x_n - y_n)^2} \quad (10)$$

The original dataset contains a concatenated matrix of feature vectors, where each column represents a sample and each row represents a feature. This matrix is used to compute a distance matrix (D) that captures the pairwise distances between all samples. The distance matrix helps understand the dissimilarity or similarity between the original data points in the high-dimensional space. The

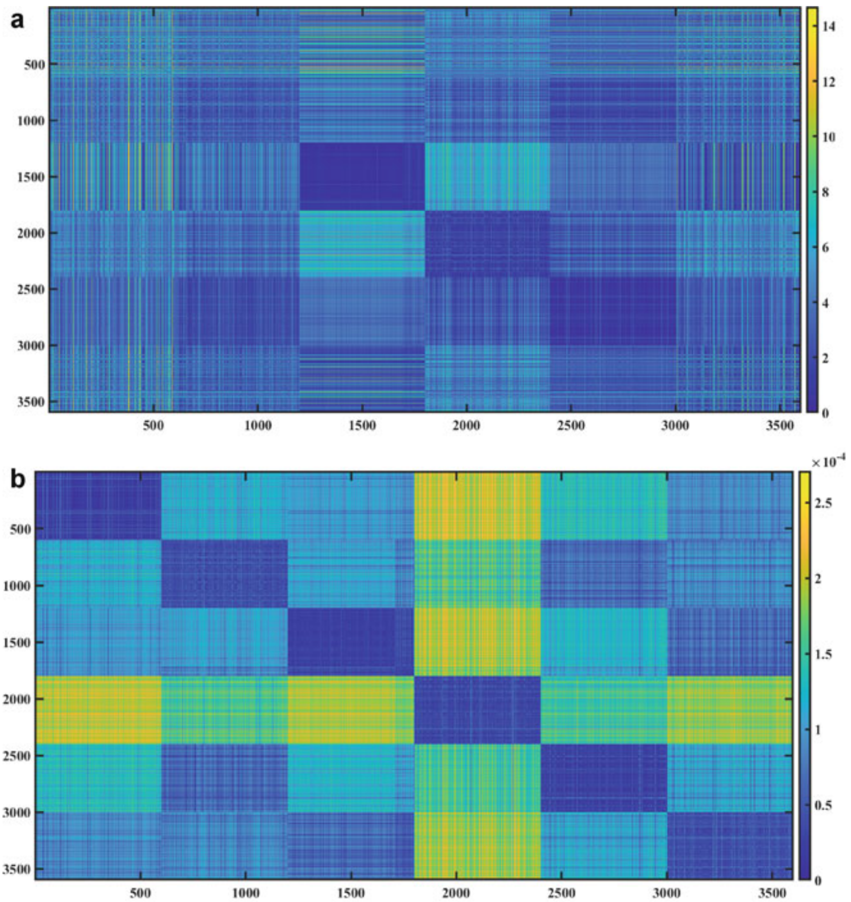


Fig. 3 Figure (**a**) The distance before applying KLDA and (**b**) the distance matrix after applying KLDA

feature vectors are projected from higher-dimension data to lower-dimensional space after performing kernel linear discriminant analysis (KLDA). It contains feature vectors where the distance matrix is computed similarly to the original data samples. The significance of computing a distance matrix from the higher-dimension data into lower-dimension space feature vectors is that it allows us to see how well the KLDA has managed to separate different data classes. Samples from the same class should be close, and samples from different classes should be farther apart. By computing the distance matrix for the kLDA-transformed feature vectors, it visualizes the extent to which the classes are linearly separable from higher-dimensional data into lower-dimensional space. KLDA effectively reduces dimensionality and improves classification performance by combining the benefits of LDA and kernel approaches.

Results and Discussion

This chapter presents the results of our novel image classification and algorithm applied to a set of test images. The Kernel linear discriminant analysis algorithm combines Gaussian kernel calculations, eigen analysis, and distance metrics to achieve the desired outcomes. A series of experiments was conducted using a dataset comprising various test images and corresponding ground truth data. The algorithm randomly selects regions from the test images and their corresponding ground truth data. A set of Gaussian kernel values is calculated for each selected region by comparing the image subblocks with a predefined set of kernel matrices. These values are assembled into a matrix for further analysis. The calculated Gaussian kernel matrix is then projected onto a set of eigenvectors derived from the data. This step reduces the dimensionality of the data while retaining essential features for classification. After projecting the Gaussian kernel matrix, the squared distances between the projected data and predefined eigenvectors are computed. The index corresponding to the eigenvector with the minimum squared distance is assigned to each data point. This process results in an index matrix representing the test image's classified regions. The obtained index matrix is visualized through a series of subplots for each test image. The first subplot displays the original test image. The subsequent subplots show the original region of interest and the predicted index matrix as a color-coded image, where different colors correspond to different classifications. The image is declared to the corresponding class group based on the majority index values predicted in the matrix (subblocks). (Refer to Figs. 4, 5, 6, 7, 8, and 9).

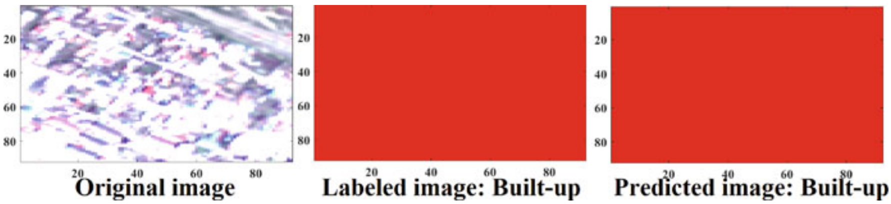


Fig. 4 Sample test image 1: Class index of the subblocks (size 9 × 9) correctly predicted is 100%

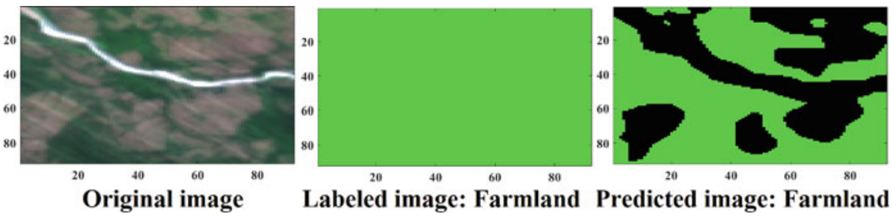


Fig. 5 Sample test image 2: Class index of the subblocks (size 9 × 9) correctly predicted is 57.2%

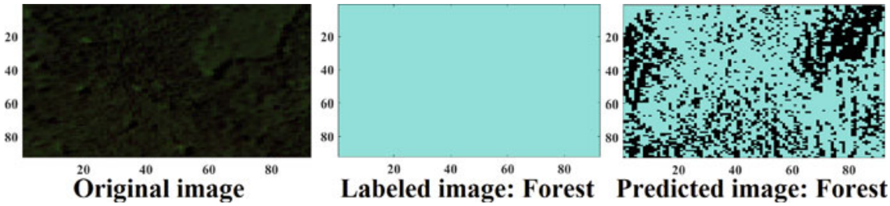


Fig. 6 Sample test image 3: Class index of the subblocks (size 9 × 9) correctly predicted is 86.3%

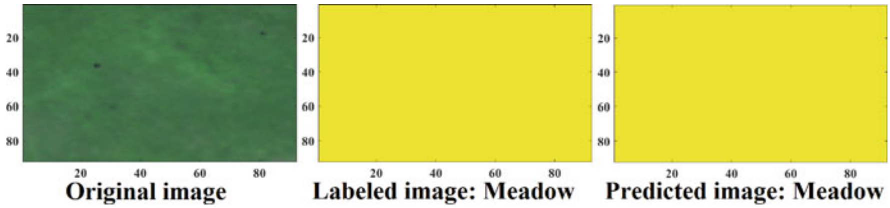


Fig. 7 Sample test image 4: Class index of the subblocks (size 9 × 9) correctly predicted is 100%

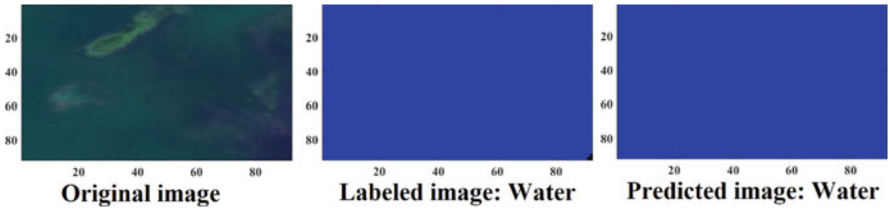


Fig. 8 Sample test image 5: Class index of the subblocks (size 9 × 9) correctly predicted is 100%

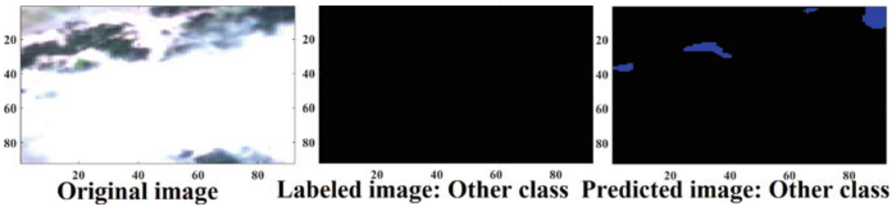


Fig. 9 Sample test image 6: Class index of the subblocks (size 9 × 9) correctly predicted is 97.76%

The proposed algorithm KLDA represents better results in image classification. The algorithm's ability to classify regions based on calculated Gaussian kernel values and projected eigenvalues demonstrates its potential for various applications such as object detection and scene segmentation. However, certain limitations exist in the current implementation. The algorithm's performance heavily depends on the choice of parameters, such as the size of the image subblocks and the number of eigenvectors used. Further optimization and parameter tuning may be required to achieve consistent and robust results across diverse datasets. The presented results highlight the algorithm's potential for region classification and acknowledge the need for further refinement and parameter optimization.

Conclusion

The proposed technique contributes to high-resolution remote sensing image analysis by introducing a novel approach that leverages the strengths of kernel linear discriminant analysis (K-LDA). The classification method, K-LDA, harnesses the advantages of dimensionality reduction and class separation. The incorporation of kernel functions enables K-LDA to efficiently handle nonlinear data distributions, making it a valuable tool for appropriate and reliable high-resolution remote sensing image classification.

The groundwork for our proposed method is laid through an in-depth exploration of the theoretical background encompassing principal component analysis (PCA) and linear discriminant analysis (LDA). The principal component analysis (PCA) demonstrated its capability in dimensionality reduction, allowing for the transformation of complex data into a more manageable space while retaining essential variance. However, its oversight of class-specific information limits its applicability in classification tasks where discriminating between distinct classes is paramount. Linear discriminant analysis (LDA) for class separability emerged as a valuable technique for classification endeavors. LDA enhances the distinction between different classes by maximizing inter-class variance and minimizing intra-class variance. Nevertheless, its linear nature restricts its adaptability to intricate nonlinear data distributions, characteristic of high-resolution remote sensing images.

The novel kernel linear discriminant analysis (K-LDA) was introduced to address these limitations. By incorporating kernel functions (Gaussian kernel), K-LDA extends LDA's capabilities into high-dimensional spaces, allowing for the capture of intricate nonlinear relationships within the data. This unique attribute positions K-LDA as a promising candidate for classifying high-resolution remote sensing images, where complex spatial patterns and diverse land covers necessitate a nuanced approach. In the evaluation phase, the proposed K-LDA method underwent rigorous testing on a benchmark dataset of high-resolution remote sensing images. By amalgamating the strengths of dimensionality reduction and class discrimination and by the utilization of kernel functions, K-LDA demonstrates efficacy in managing complex, nonlinear data distributions. As such, K-LDA is a valuable asset in pursuing accurate, reliable, and robust high-resolution remote sensing image classification – an advancement with broad implications across various domains, including land cover mapping, urban planning, and environmental monitoring.

Future Scope

There are several avenues for future research and exploration. Investigating the impact of different kernel functions in KLDA could provide a deeper understanding of their effectiveness in class separability. Furthermore, the proposed methodology can be extended to tackle more complex datasets with more classes or instances. Evaluating the scalability of the approach on larger datasets and assessing its computational efficiency will be critical for real-world applications. Conducting comparative studies with other state-of-the-art classification methods would help validate the competitiveness and generalizability of our approach across various datasets and domains. Overall, this chapter lays the groundwork for future research endeavors in feature extraction and classification techniques, offering new opportunities for advancing the field of machine learning and its practical applications.

References

Ara Z, Jha R, Quaff AR (2023) Critical appraisal of satellite data for land use/land cover classification and change detection: a review. In: International Conference on Hydraulics, Water Resources and Coastal Engineering. https://doi.org/10.1007/978-981-99-1901-7_11

Christopher MB (2006) Pattern recognition and machine learning. In: Information science and statistics. Springer, New York, NY

Gongjin L, Zhenyu G, Lingyao T, Ting L (2022) Class binarization to neuroevolution for multiclass classification. Neural Comput Appl. https://doi.org/10.1007/s00521-022-07525-6

Gopi ES (2020) Pattern recognition and computational intelligence techniques using Matlab. In: Transactions on computational science and computational intelligence (TRACOSCI), Springer Nature Switzerland AG

Guo Y, Yin X, Zhao X, Yang D, Bai Y (2019) Hyperspectral image classification with SVM and guided filter. EURASIP J Wireless Commun Netw. https://doi.org/10.1186/s13638-019-1346-z

Hasanlou M, Samadzadegan F, Homayouni S (2015) SVM-based hyperspectral image classification using intrinsic dimension. Arab J Geosci. https://doi.org/10.1007/s12517-013-1141-9

Hidalgo DR, Cortés BB, Bravo EC (2015) Dimensionality reduction of hyperspectral images of vegetation and crops based on self-organized maps. Inf Process Agric. https://doi.org/10.1016/j.inpa.2020.07.002

Jan B, Rahman O, Parveen S, Khan SA (2023) Multi-stage binary classification technique for incipient fault diagnosis of oil immersed power transformers based on ANFIS. In: International Conference on Power, Instrumentation, Energy and Control (PIECON). https://doi.org/10.1109/PIECON56912.2023.10085900

JayaBrindha G, Subbu ESG (2018) Ant colony technique for optimizing the order of cascaded SVM classifier for sunflower seed classification. IEEE Trans Emerging Top Comput Intell. https://doi.org/10.1109/TETCI.2017.2772918

Jia W, Sun M, Lian J, Hou S (2022) Feature dimensionality reduction: a review. Complex Intell Syst. https://doi.org/10.1007/s40747-021-00637-x

Ruoxi L, Yinong Z, Xuedong Z, Yuhan C, Xiaojing C (2020) Banknote infrared characteristics discrimination based on PCA and LDA. In: Communications in computer and information science book series. https://doi.org/10.1007/978-981-15-8086-4_47

Li H, Cui J, Zhang X, Han Y, Cao L (2022) Dimensionality reduction and classification of hyperspectral remote sensing image feature extraction. Remote Sens. https://doi.org/10.3390/rs14184579

Liu J, Chen S (2005) Resampling LDA/QR and PCA+LDA for face recognition. Lecture notes in computer science. https://doi.org/10.1007/11589990_174

Liu J, Zhang T, Liu Y, Zhao F (2022) Hyperspectral remote sensing images feature extraction based on higher-order spectra. IEEE Geosci Remote Sensing Lett. https://doi.org/10.1109/LGRS.2022.3206917

Martinez AM, Kak AC (2001) PCA versus LDA. IEEE Trans Pattern Anal Mach Intell. https://doi.org/10.1109/34.908974

Otani N, Otsubo Y, Koike T, Sugiyama M (2020) Binary classification with ambiguous training data. In: Machine Learning. Springer. https://doi.org/10.1007/s10994-020-05915-2

Raghavendra U, Mahesh PK, Gudigar A (2012) A novel face recognition method using PCA, LDA and support vector machine. Lecture notes of the institute for computer sciences, Social informatics and telecommunications engineering. https://doi.org/10.1007/978-3-642-27308-7_25

Rajeswara Rao D, Noorjahan S, Ayesha Fathima S (2022) Classification of land cover usage from satellite images using deep learning algorithms. In: 2022 International Conference on Electronics and Renewable Systems (ICEARS). https://doi.org/10.1109/ICEARS53579.2022.9752282

Sadeghi MT, Samiei M, Kittler J (2001) Fusion of PCA-based and LDA-based similarity measures for face verification. EURASIP J Adv Signal Process. https://doi.org/10.1155/2010/647597

Santhi N, Annbuselvi K, Sivakumar S (2022) An efficient gabor scale average (GSA) based PCA to LDA feature extraction of face and gait cues for multimodal classifier. In: Innovations in computational intelligence and computer vision. https://doi.org/10.1007/978-981-19-0475-2_14

Shemul MSH, Rahman MM, Ahmed S, Marjan MA, Uddin MP, Afjal MI (2022) Segmented-sparse-PCA for hyperspectral image classification. In: 4th International Conference on Electrical, Computer and Telecommunication Engineering (ICECTE). https://doi.org/10.1109/ICECTE57896.2022.10114537

Tsoulfidis L, Athanasiadis I (2022) A new method of identifying key industries: a principal component analysis. J Econ Struc. https://doi.org/10.1186/s40008-022-00261-z

Varma S, Shinde M, Chavan SS (2019) Analysis of PCA and LDA features for facial expression recognition using SVM and HMM classifiers. Techno-Societal 2018. https://doi.org/10.1007/978-3-030-16848-3_11

Wang L, Chan KL, Xue P, Zhou L (2008) A Kernel-induced space selection approach to model selection in KLDA. IEEE Trans Neural Netw. https://doi.org/10.1109/TNN.2008.2005140

Xiaoming Z, Shiqing Z (2012) Facial expression recognition using local binary patterns and discriminant kernel locally linear embedding. EURASIP J Adv Signal Process. https://doi.org/10.1186/1687-6180-2012-20

Xin-Yi T, Gui-Song X, Qikai L, Huanfeng S, Shengyang L, Shucheng Y, Liangpei Z (2020) Land-cover classification with high-resolution remote sensing images using transferable deep models. In: Remote sensing of environment. https://doi.org/10.1016/j.rse.2019.111322

Xin-Yi T, Gui-Song X, Qikai L, Huanfeng S, Shengyang L, Shucheng Y, Liangpei Z (2020) Land-cover classification with high-resolution remote sensing images using transferable deep models. R Soc Chem. (Available via open source license). https://x-ytong.github.io/project/GID.html