ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 10, Issue 5, May 2023

Comparison of Numerous Feature Extraction Methods in Face Recognition

[1] Harshit Mittal, [2] Dr. Neeraj Garg

[1] [2] Maharaja Agrasen Institute of Technology, Rohini, Delhi, India
Corresponding Author Email: [1] mittalharshit99@gmail.com

Abstract— ***This paper compares different feature extraction methods with different classification techniques of face recognition technology. The author starts by discussing the various steps involved in a typical face recognition system, including data visualization, feature extraction, training, and testing the model. After visualizing the dataset, the author then delves into the various feature extraction methods used in face recognition, including Principal Component Analysis (PCA), Independent Component Analysis (ICA), Locally Linear Embedding (LLE), Local Binary Pattern(LBP), and Simple Autoencoder. The author compares the performance of the above feature extraction methods by training the models using Support Vector Clustering(SVC), and Linear Discriminant Analysis(LDA) algorithms. The results of the experiments help in comparing various feature extraction methods and finding the best feature extraction method for an efficient face recognition system. The authors conclude by discussing the potential applications of face recognition technology, including security systems, biometrics, and human-computer interaction. They highlight the growing importance of face recognition in various domains, including law enforcement, healthcare, and entertainment, and emphasize the need for further research and development in this field. The paper provides valuable insights into the current state-of-the-art in face recognition technology and will be of interest to researchers, engineers, and practitioners working in the field of computer vision and pattern recognition.***

Keywords: Face Recognition, Feature Extraction, Principal Component Analysis (PCA), Independent Component Analysis (ICA), Locally Linear Embedding (LLE), Local Binary Pattern(LBP), Simple Autoencoder, Support Vector Clustering(SVC), Linear Discriminant Analysis(LDA).

I. INTRODUCTION

First Let us understand what is computer vision under which the technology of face recognition lies. Computer Vision[1] is a field of study within computer science and engineering that focuses on enabling computers to interpret and understand visual information in the same way as humans. It involves techniques for acquiring, processing, analyzing, and understanding images and videos to extract meaningful information and make decisions based on that information. Computer vision applications include image recognition, object detection, tracking, facial recognition, medical imaging, and much more.

This paper is based on face recognition[2,3] which is a subfield of computer vision that identifies and verifies individuals based on their facial features. It uses algorithms to analyze an image or video of a face and compare it to a database of known faces to determine the identity of the person. The process involves detecting the face in the image, extracting unique features such as the shape of the eyes, nose, and mouth, and comparing those features to a database to find a match. Face recognition technology has numerous applications, including security systems, photo tagging, and mobile device unlocking.

In this paper, the authors compare the various feature extraction methodologies in the order of Principal Component Analysis (PCA), Independent Component Analysis (ICA), Locally Linear Embedding (LLE), Local Binary Pattern(LBP), and then, Simple Autoencoder using Olivetti dataset. The author uses Support Vector Clustering(SVC), and Linear Discriminant Analysis(LDA) for training the above models separately and compares their accuracy. At last, in the result and discussions, the authors also enlist the use of each extraction method for the provided accuracy.

The paper is further divided into different sections, Data Visualization, Feature Extraction Methods, Comparison Analysis, and Result and Discussions.

II. DATA VISUALIZATION

The Olivetti faces dataset is a renowned dataset and acts as a base dataset for numerous facial recognition models, it was formed at AT&T Laboratories established in Cambridge, UK. The dataset is widely used in academic research and has become a standard benchmark for testing the performance of machine learning algorithms in the field of computer vision. It is available on Kaggle as a public data set for use in machine learning and data science projects.

The Olivetti dataset contains ten distinct face images of 40 distinct people each taken between April 1992 and April 1994. In the above dataset, Face images were taken at different times under varying lightening and facial expressions with a black background at a Gray level.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 10, Issue 5, May 2023

```
In [4]:
print("There are {} images in the dataset".format(len(data)))
print("There are {} unique targets in the dataset".format(len(np.unique(target))))
print("Size of each image is {}x{}".format(data.shape[1],data.shape[2]))
print("Pixel values were scaled to [0,1] interval. e.g:{}".format(data[0][0,:4]))
```

```
There are 400 images in the dataset
There are 40 unique targets in the dataset
Size of each image is 64x64
Pixel values were scaled to [0,1] interval. e.g:[0.30991736 0.3677686  0.41735536 0.44214877]
```

```
In [5]:
print("unique target number:",np.unique(target))
```

```
unique target number: [ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39]
```

Fig.1 Overview of Dataset [courtesy: self]

In the above image, the overview of the whole structure of the dataset is shown.

```
In [7]:
show_40_distinct_people(data, np.unique(target))
```

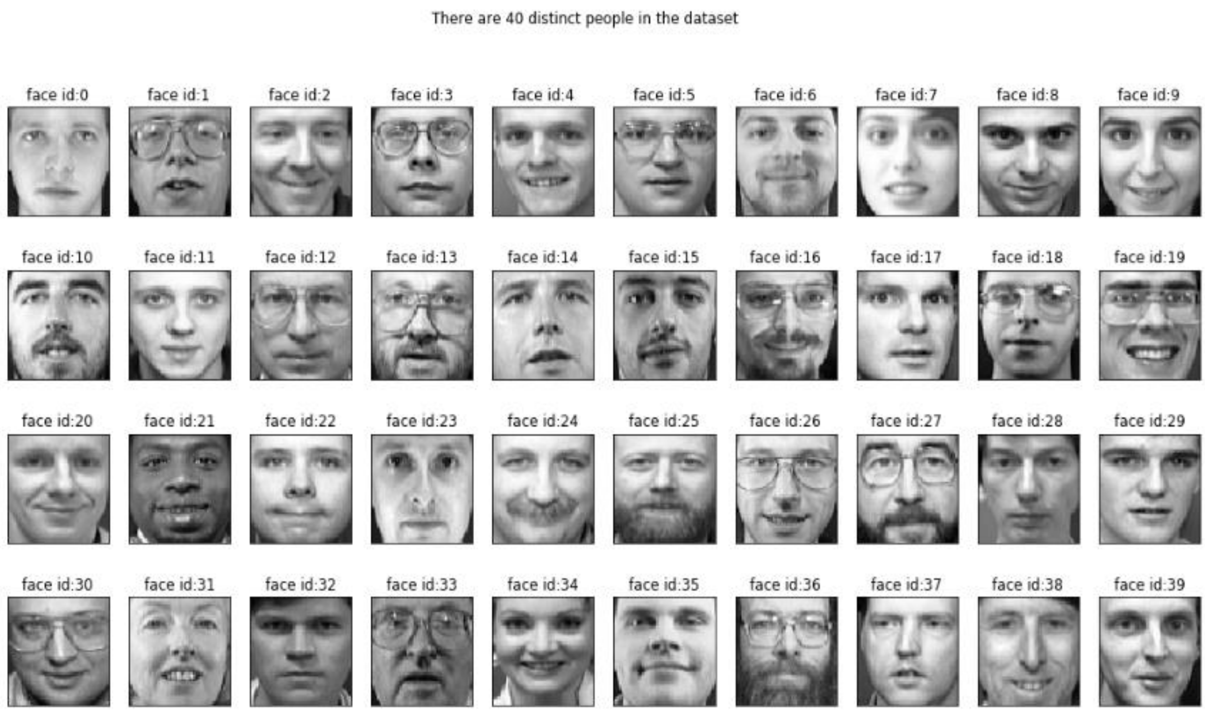

Fig.2 40 people used in the dataset [courtesy: self]

In the above image, each of the 40 people is shown with each one of their images.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 10, Issue 5, May 2023

```
In [9]:
#You can playaround subject_ids to see other people faces
show_10_faces_of_n_subject(images=data, subject_ids=[0,5, 21, 24, 36])
```

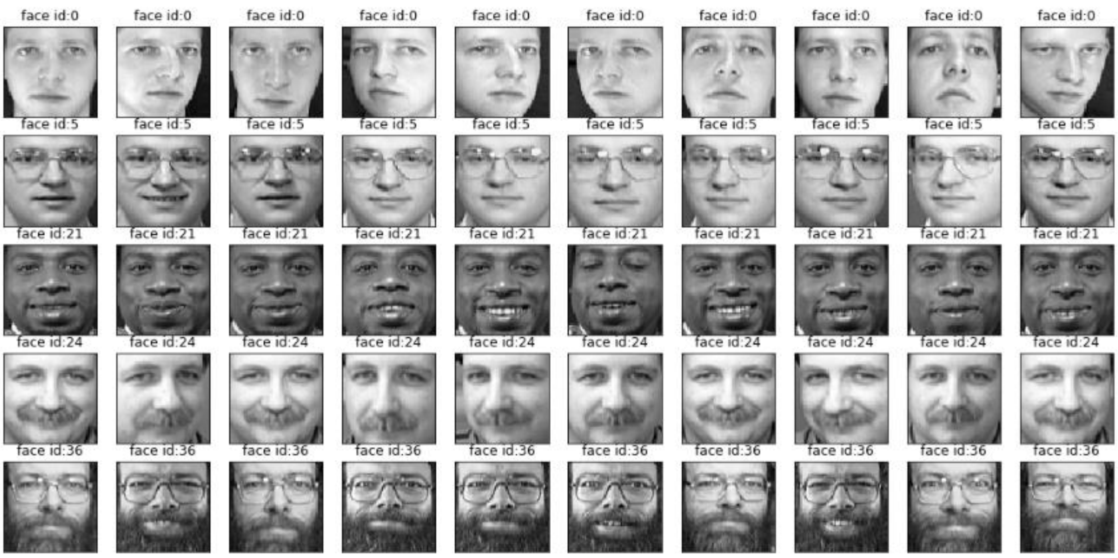

Fig.3 Same people with different expressions [courtesy: self]

In the above image, different 10 images of each of 5 different people are shown. Each image of the same person with different lighting and facial expressions.

III. FEATURE EXTRACTION METHODS

This section consists of the main methodology of this paper. In this section, the authors compare the performance of numerous feature extraction methods[3] by training the models using Support Vector Clustering(SVC), and Linear Discriminant Analysis(LDA) algorithms.

3.1. Principal Component Analysis (PCA)

Principal Component Analysis (PCA) [3,5] is a widely used technique in computer vision and machine learning, including face recognition. The main idea behind PCA is to transform the original high-dimensional data into a lower-dimensional representation while preserving as much information as possible. This can be useful in face recognition because images of faces can contain a large number of features, such as the shape of the nose, the size of the eyes, the position of the mouth, etc. However, not all these features are equally important for face recognition, and some of them might even be irrelevant.

PCA can help identify an image's most important features, known as the "principal components." These principal components correspond to the directions in the feature space that have the maximum variance. By projecting the original images onto these principal components, we can obtain a lower-dimensional representation of the data that still captures the main variations in the faces.

Once the PCA transformation is obtained, it can be used to perform face recognition by comparing the PCA representations of two face images.

```
clf = SVC()
clf.fit(X_train_pca, y_train)
y_pred = clf.predict(X_test_pca)
print("accuracy score:{:.6f}".format(metrics.accuracy_score(y_test, y_pred)))

accuracy score:0.916667
```

```
lr=LinearDiscriminantAnalysis()
lr.fit(X_train_pca, y_train)
y_pred=lr.predict(X_test_pca)
print("Accuracy score:{:.6f}".format(metrics.accuracy_score(y_test, y_pred)))

Accuracy score:0.925000
```

Fig.4 Accuracy for PCA model [courtesy: self]

ISSN (Online) 2394-2320
International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 10, Issue 5, May 2023

3.2. Independent Component Analysis (ICA)

Independent Component Analysis (ICA) [3,6] is a technique in machine learning that can be used for face recognition and other applications. Like Principal Component Analysis (PCA), ICA is a method for dimensionality reduction, but it operates on a different principle. While PCA seeks to identify the directions in the feature space with the maximum variance, ICA aims to identify the directions in the feature space that are statistically independent.

In face recognition, ICA can be used to extract a set of features from the face images that are independent of each other. These features can then be used to represent the face images in a lower-dimensional space. By doing this, ICA can help to reduce the noise and variability in the data, which can make face recognition more robust and accurate.

The ICA transformation can be used in a similar way to the PCA transformation for face recognition. For example, one common approach is to use the Euclidean distance between the ICA representations of two face images as a measure of similarity. If the distance is below a certain threshold, the two faces are considered to be the same person.

```
clf = SVC()
clf.fit(X_train_ica, y_train)
y_pred = clf.predict(X_test_ica)
print("accuracy score:{:.6f}".format(metrics.accuracy_score(y_test, y_pred)))
```

```
accuracy score:0.916667
```

```
lr=LinearDiscriminantAnalysis()
lr.fit(X_train_ica, y_train)
y_pred=lr.predict(X_test_ica)
print("Accuracy score:{:.6f}".format(metrics.accuracy_score(y_test, y_pred)))
```

```
Accuracy score:0.933333
```

Fig.5 Accuracy for ICA model [courtesy: self]

3.3. Locally Linear Embedding (LLE)

Locally Linear Embedding (LLE) [3] is a technique in machine learning that can be used for face recognition and other applications. Like PCA and ICA, LLE is a method for dimensionality reduction, but it operates on a different principle. While PCA and ICA seek to identify global patterns in the data, LLE seeks to preserve local relationships between the data points.

In face recognition, LLE can be used to extract a set of features from the face images that preserve the local relationships between the pixels in the images. This can help to reduce the noise and variability in the data, making face recognition more robust and accurate.

The LLE transformation can be used in a similar way to the PCA and ICA transformations for face recognition. For example, one common approach is to use the Euclidean distance between the LLE representations of two face images as a measure of similarity. If the distance is below a certain threshold, the two faces are considered to be the same person.

```
clf = SVC()
clf.fit(X_train_lle, y_train)
y_pred = clf.predict(X_test_lle)
print("accuracy score:{:.6f}".format(metrics.accuracy_score(y_test, y_pred)))
```

```
accuracy score:0.900000
```

```
lr=LinearDiscriminantAnalysis()
lr.fit(X_train_lle, y_train)
y_pred=lr.predict(X_test_lle)
print("Accuracy score:{:.6f}".format(metrics.accuracy_score(y_test, y_pred)))
```

```
Accuracy score:0.908333
```

Fig.6 Accuracy for LLE model [courtesy: self]

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 10, Issue 5, May 2023

3.4. Local Binary Pattern (LBP)

Local Binary Patterns (LBP) [3,7] is a texture-based feature extraction method that can be used in computer vision and machine learning, including face recognition. The main idea behind LBP is to capture the local structure of an image by comparing the intensity values of a pixel to its neighbors.

In face recognition, LBP can be used to extract features from the face images that capture the local structure of the facial features, such as the eyes, nose, and mouth. The LBP features can be calculated for each pixel in the image, and the resulting feature vectors can be used to represent the face images in a lower-dimensional space.

Once the LBP features are obtained, they can be used for face recognition in a similar way to other feature extraction methods, such as PCA, ICA, and LLE. For example, one common approach is to use a classifier, such as a support vector machine (SVM) or a k-nearest neighbors (k-NN) algorithm, to compare the LBP features of two face images and determine if they belong to the same person.

```
clf = SVC()
clf.fit(lbp_1, y_train)
y_pred = clf.predict(lbp_2)
print("accuracy score:{:.6f}".format(metrics.accuracy_score(y_test, y_pred)))
```

```
accuracy score:0.608333
```

```
lr=LinearDiscriminantAnalysis()
lr.fit(lbp_1, y_train)
y_pred=lr.predict(lbp_2)
print("Accuracy score:{:.6f}".format(metrics.accuracy_score(y_test, y_pred)))
```

```
Accuracy score:0.900000
```

Fig.7 Accuracy for LBP model [courtesy: self]

3.5. Simple Autoencoder

An Autoencoder[3,8] is a type of neural network that can be used for various tasks in computer vision, including face recognition. In its simplest form, an autoencoder is an unsupervised learning model that aims to reconstruct its input, given a lower-dimensional representation of the data.

In face recognition, an autoencoder can be used to extract features from the face images that capture the most important information for recognizing a face. The autoencoder is trained to reconstruct the original face images from a lower-dimensional representation, known as encoding. This encoding can be thought of as a compact representation of the face image that captures the most important features for recognition.

Once the autoencoder is trained, the encoding can be used for face recognition. For example, one common approach is to use the Euclidean distance between the encodings of two face images as a measure of similarity. If the distance is below a certain threshold, the two faces are considered to be the same person.

```
clf = SVC()
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
print("accuracy score:{:.5f}".format(metrics.accuracy_score(y_test, y_pred)))
```

```
accuracy score:0.88333
```

```
lr=LinearDiscriminantAnalysis()
lr.fit(X_train, y_train)
y_pred=lr.predict(X_test)
print("Accuracy score:{:.5f}".format(metrics.accuracy_score(y_test, y_pred)))
```

```
Accuracy score:0.94167
```

Fig.8 Accuracy for simple autoencoder model [courtesy: self]

IFERP
connecting engineers... developing research

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 10, Issue 5, May 2023

IV. COMPARISON ANALYSIS

Before the comparison of feature extraction models, let's grab some knowledge about base models Support Vector Clustering(SVC), and Linear Discriminant Analysis(LDA).

Support Vector Clustering (SVC)[9] and Linear Discriminant Analysis (LDA)[10] are two different machine-learning techniques that can be used for face recognition.

SVC is a clustering algorithm that partitions the data into clusters such that each cluster is as homogeneous as possible and the separation between the clusters is maximized. In face recognition, SVC can be used to group face images into different classes, such as the images of different individuals. Given a set of face images, SVC seeks to find a decision boundary that separates the images into distinct classes. This boundary is represented by a subset of the data points, known as support vectors, that determine the location and shape of the boundary.

LDA, on the other hand, is a dimensionality reduction technique that transforms the data into a lower-dimensional space while preserving class separability. In face recognition, LDA can be used to extract features from the face images that capture the most important information for recognizing a face. The goal of LDA is to find a projection of the data that maximizes the class separability, such that the projected data is well-suited for classification.

Once the features are obtained using SVC or LDA, they can be similarly used for face recognition. For example, one common approach is to use a classifier, such as a support vector machine (SVM) or a k-nearest neighbors (k-NN) algorithm, to compare the features of two face images and determine if they belong to the same person.

In summary, both SVC and LDA are useful techniques for face recognition, and they can be used in combination to enhance the performance of face recognition systems. SVC provides a way to group face images into different classes, while LDA transforms the data into a lower-dimensional space that is well-suited for classification. SVC and LDA can be used in combination with other techniques, such as PCA, ICA, LLE, LBP, and Autoencoders, to enhance the performance of face recognition systems.

Below is the table for an accuracy comparison of each feature extraction method discussed above trained using Support Vector Clustering(SVC), and Linear Discriminant Analysis(LDA).

Table 1. Accuracy of Feature Extraction Methods

Feature Extraction Method	Accuracy with SVC	Accuracy with LDA
Principal Component Analysis(PCA)	0.9167	0.9250
Independent Component Analysis(ICA)	0.9167	0.9333
Locally Linear Embedding (LLE)	0.9000	0.9083
Local Binary Pattern(LBP)	0.6083	0.9000
Simple Autoencoder	0.8833	0.9417

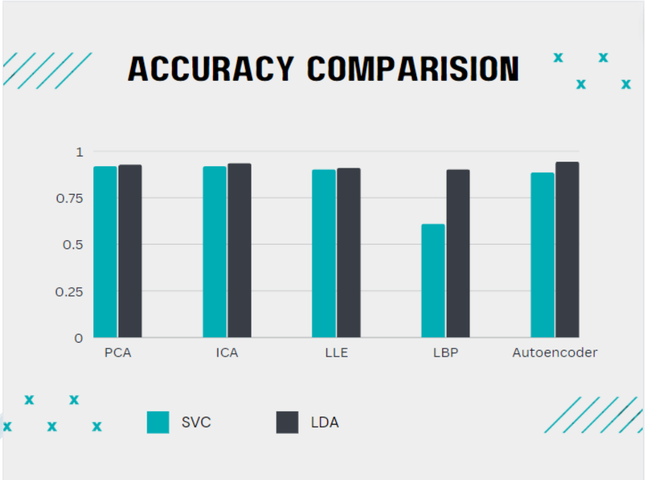

Fig.9 Accuracy for simple autoencoder model [courtesy: self]

V. RESULT AND DISCUSSIONS

Well, from the above comparison analysis and Table 1, one can assume that for Support Vector Clustering(SVC) both Principal Component Analysis(PCA) and Independent Component Analysis(ICA) showed the highest accuracy of 91.67%. Whereas, in Linear Discriminant Analysis(LDA) Simple Autoencoder shows the highest accuracy of 94.17% and Independent Component Analysis(ICA) shows the second highest accuracy of 93.33%. In general, we can see that ICA and PCA show a great average of results which is true for the above dataset and face recognition technology whereas it could be different for some other datasets and any other use of these feature extraction methods.

VI. CONCLUSION

In conclusion, the authors found that Independent Component Analysis(ICA) and Principal Component Analysis(PCA) are the two feature extraction methods that can be used for good accuracy models. Well, in recent times face recognition technology has made remarkable advancements in recent years. The ability to accurately identify individuals based on their facial features has numerous potential applications in various fields, such as security, marketing, and entertainment. Despite its growing popularity, the technology still faces significant challenges, including privacy concerns and difficulties in recognizing individuals from diverse backgrounds.

Therefore, researchers and practitioners must continue exploring and improving upon the existing face recognition methods and algorithms. This will involve not only refining the accuracy and speed of recognition but also addressing privacy and ethical concerns by developing secure and transparent systems. As technology evolves and becomes more sophisticated, it will be essential to strike a balance between its benefits and potential risks to ensure that it is used ethically and responsibly.

VII. DECLARATIONS

Figures are screenshots from the code of the authors.

Consent to participate: both authors agreed to participate and publishing of the paper.

Consent for publication: both authors agreed for publishing the paper.

Code availability: can be provided on demand or could be checked in the GitHub repository on ‘TheHarshitMittal’.

Ethical Approval: All the data and research are done via the consent of referenced authors.

Availability of Supporting Data: No new data is created.

Competing Interests: Not Applicable.

Funding: Not Applicable.

Author’s contributions: Code, comparison of features, and literature of the paper are done by Harshit Mittal and Dr. Neeraj Garg.

Acknowledgments: The author thanks all the referenced authors for the data of the above research.

REFERENCES

[1] Computer vision. (2023, February 1). In Wikipedia. https://en.wikipedia.org/wiki/Computer_vision
[2] Facial recognition system. (2023, February 11). In Wikipedia. https://en.wikipedia.org/wiki/Facial_recognition_system
[3] Mittal, H., & Garg, N. (2023). Recognizing/Detecting Human Faces in Images: Survey. Available at SSRN 4345630.
[4] Hao, J., & Ho, T. K. (2019). Machine learning made easy: a review of scikit-learn package in python programming language. Journal of Educational and Behavioral Statistics, 44(3), 348-361.
[5] Chen, J., & Jenkins, W. K. (2017, August). Facial recognition with PCA and machine learning methods. In 2017 IEEE 60th international Midwest symposium on circuits and systems (MWSCAS) (pp. 973-976). IEEE.
[6] Annamalai, P., Raju, K., & Ranganayakulu, D. (2018). Soft Biometrics Traits for Continuous Authentication in Online Exam Using ICA Based Facial Recognition. Int. J. Netw. Secur., 20(3), 423-432.
[7] Shetty, A. B., & Rebeiro, J. (2021). Facial recognition using Haar cascade and LBP classifiers. Global Transitions Proceedings, 2(2), 330-335.
[8] Finizola, J. S., Targino, J. M., Teodoro, F. G., & Lima, C. A. (2019, July). Comparative study between deep face, autoencoder and traditional machine learning techniques aiming at biometric facial recognition. In 2019 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE.
[9] Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., & Lopez, A. (2020). A comprehensive survey on support vector machine classification: Applications, challenges and trends. Neurocomputing, 408, 189-215.
[10] Zhu, F., Gao, J., Yang, J., & Ye, N. (2022). Neighborhood linear discriminant analysis. Pattern Recognition, 123, 108422.