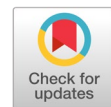


Gender classification performance optimization based on facial images using LBG-VQ and MB-LBP



Faruq Abdul Hakim ^{a,1}, Tio Dharmawan ^{a,2,*}, Muhamad Arief Hidayat ^{a,3}

^a University of Jember, Jl. Kalimantan Tegalboto No.37, Krajan Timur, Sumbersari, Kec. Sumbersari, Kabupaten Jember, Jawa Timur, Indonesia

¹ faruqabduhkim2001@gmail.com; ² tio.pssi@unej.ac.id; ³ arief.hidayat@unej.ac.id

* corresponding author

ARTICLE INFO

Article history

Received November 15, 2024

Revised February 5, 2025

Accepted February 6, 2025

Available online February 28, 2025

Keywords

Feature extraction

Machine learning

Gender classification

Linde buzo gray vector quantization

Multi-block local binary pattern

ABSTRACT

In the computer vision and machine learning field, especially for gender classification based on facial images, feature extraction is one of the inseparable parts. Various features can be extracted from images, including texture features. Several prior studies show that the Linde Buzo gray vector quantization (LBG-VQ) and Multi-block local binary pattern (MB-LBP) methods can extract texture features from images. The LBG-VQ produces less optimal performance in gender classification on the FEI facial images dataset. On the other hand, the MB-LBP produces more optimal performance when applied to the FERET facial images dataset. Therefore, this study was conducted to discover the gender classification performance when the LBG-VQ and MB-LBP methods are implemented independently or in combination on the FEI facial images dataset. Three preprocessing stages are involved before extracting images' features: noise removal, illumination adjustment, and image conversion from RGB to grayscale. The extracted features are then used as training material for several classification methods, namely Naïve Bayes, SVM, KNN, Random Forest, and Logistic Regression. Then, the K-Fold Cross Validation method is used to evaluate the trained models. This study discovered that the implementation of MB-LBP tends to show a performance improvement compared to the LBG-VQ. Furthermore, the most optimal classification model, with a performance of 91.928%, was formed by implementing Logistic Regression with MB-LBP on LBG-VQ quantized images. In conclusion, this study successfully formed an optimized gender classification model based on the FEI facial images dataset.



© 2025 The Author(s).

This is an open access article under the [CC-BY-SA](#) license.



1. Introduction

Faces are a reliable metric for identifying humans because every human being has a unique face [1]. While humans can easily differentiate between genders based on perception, machines face a more complex challenge in gender classification due to their lack of human-like perception. Machines classify gender based on facial features and algorithms [2]. Machines use inputs such as images, and feature extraction from images is one of the critical stages. Various features can be extracted from an image, including pixel values, color histograms, textures, shapes, and so on. Some prior studies have been conducted related to gender classification based on facial images. In 2018, a study utilized the Linde Buzo Gray Vector Quantization (LBG-VQ) to extract features from images [2]. In the same year, another study utilized the Multi-Block Local Binary Pattern (MB-LBP) feature extraction method [3].

Linde Buzo Gray Vector Quantization (LBG-VQ) is a method for extracting texture features from images. Initially designed for data compression, LBG-VQ uses the Linde, Buzo, and Gray (LBG) algorithm for Vector Quantization (VQ) [4], [5]. VQ consists of three procedures: codebook design,

image encoding, and image decoding. The codebook design produces a codebook that represents the texture features of an image. A codebook is obtained by dividing an image into N clusters, where N determines the size of the resulting codebook. Each cluster has one centroid, which represents one texture feature of the image [2]. In 2018, a study utilized the LBG-VQ feature extraction method for gender classification based on the FEI facial images dataset [2]. The preprocessing stages involved noise removal, illumination adjustment, and RGB to gray conversion. The extracted features were used to train various machine-learning models. The study found that gender classification using codebook sizes 16 and 32 had less optimal performance, with the highest performance being 69.8%.

Apart from LBG-VQ, Multi Block Local Binary Pattern (MB-LBP), which is also utilized to extract texture features from images, is an extension of the Local Binary Pattern (LBP) method that was developed to address its shortcomings [6]. While LBP classifies textures based on single pixels and offers robustness to illumination variations, discriminative power, and simplicity [6], [7], MB-LBP uses any sub-block size of the image, providing better texture description [3], [6]. This is reinforced by a prior study in 2018 that applied the MB-LBP method for gender classification based on the FERET facial image dataset, achieving 94.7% accuracy, compared to 90.45% with the LBP method [3]. Feature extraction methods can be utilized independently or in combination. In 2018, a study combined the Improved Affinity Propagation (IAP) algorithm with the LBG algorithm to design a codebook, improving its quality compared to the conventional LBG algorithm [8]. In the same year, another study carried out the MB-LBP method for feature extraction and represented the results as a histogram [9]. These studies demonstrate that feature extraction methods can be combined.

Based on the previous studies, there was a notable gap in the literature regarding the comparative performance of those feature extraction methods. LBG-VQ applied to the FEI dataset got a performance of 69.8%, meanwhile, MB-LBP applied to the FERET dataset got a better performance of 94.7%. While individual studies have evaluated those methods separately, there was a lack of research comparing their effectiveness when applied to the same dataset. Addressing this gap is crucial because understanding the comparative performance of those methods on the same dataset can provide into their relative strength and weaknesses. This study aims to fill this gap by utilizing the FEI facial images dataset to compare the performance of LBG-VQ and MB-LBP feature extraction methods both independently and in combination. By investigating those methods on the same dataset, this study provides a clearer understanding of their effectiveness and contributes to the development of gender classification models.

The rest of this paper is structured as follows. Section 2 presents the proposed methodology. The results and discussion are presented in Section 3. Finally, the conclusion is presented in Section 4. At the end of this paper, acknowledgment, declarations, and references are presented.

2. Method

This section presents the proposed method: dataset acquisition, preprocessing, feature extraction, modeling, and evaluation. In short, the proposed method is summarized in Fig. 1.

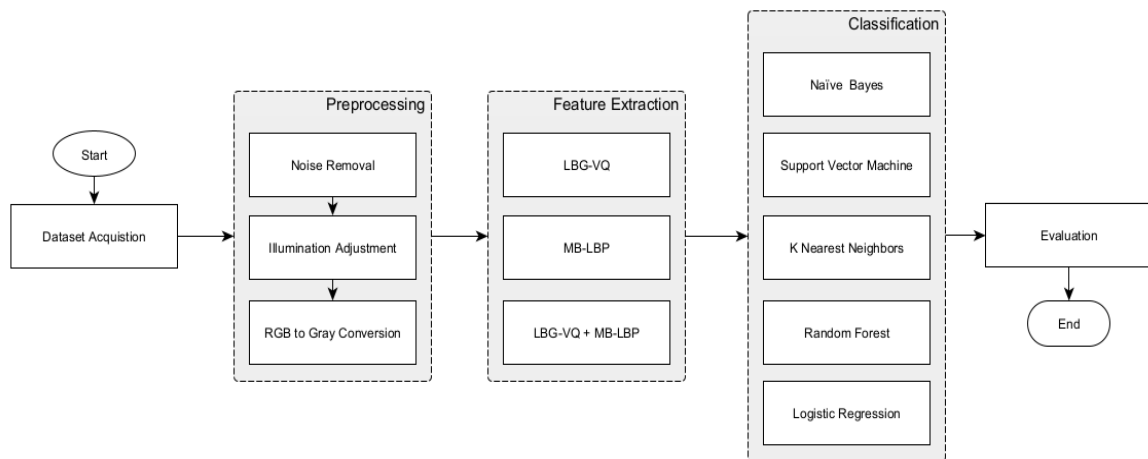


Fig. 1. Proposed Method

2.1. Dataset Acquisition

The dataset used in this study is a subset of the FEI facial images dataset [10], consisting of 200 individuals (100 men and 100 women). Each individual has two images: one smiling and one neutral. Thus, the total images used were 400 images. All images are in RGB format with faces facing forward and have a uniform size of 360x260 pixels. Examples can be seen in Fig. 2.



Fig. 2. Example of the FEI facial images dataset

2.2. Preprocessing

2.2.1. Noise Removal

This step aims to improve image quality by reducing noises by utilizing the median filter [11]–[13]. The median filter effectively denoises images with salt and pepper noise without blurring the image [12]. The basic operation involves determining the pixel's surrounding neighborhood (e.g., a 3x3 window if the kernel size set to 3), analyzing the pixel values within that neighborhood, and replacing the original pixel value with the median value [11], [13]. Fig. 3 illustrates the median filter calculation.

	110	152	168	123	139
	129	160	141	157	114
	147	112	115	118	132
	165	117	169	125	136
	130	143	109	121	154

Neighborhood values:
160, 141, 157, 112, 115,
118, 117, 169, 125

Median value: 125

Fig. 3. Median filter calculation

In this study, a median filter with kernel size set to 3 was applied to eliminate noises from the images. Fig. 4 shows an example of before and after the noise removal step, along with the difference spots.

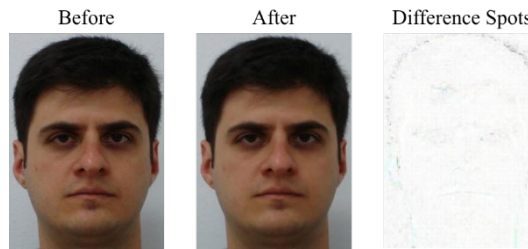


Fig. 4. Noise removal

2.2.2. Illumination Adjustment

The illumination adjustment step aims to normalize pixel intensity, ensuring uniform illumination throughout the image. This was done by applying histogram equalization [14]–[17] method, a classical technique to enhance the visual aspects of an image. It equalizes the distribution of the probability of occurrence of intensity values, allowing the result to use the full range of possible gray value [15]–[17]. The probability of a pixel with an intensity value k is formulated in Equation (1).

$$P_k = n_k / N \quad (1)$$

where n_k is the number of occurrences of the k -th intensity level within a range from 0 to $L-1$ (i.e., in 8-bit, where L is 256, ranging from 0 to 255). Meanwhile, N is the total number of pixels in an image, calculated by multiplying the number of rows and columns of pixels. The Cumulative Distribution Function (CDF) [16] is utilized to obtain the new intensity value of k -th intensity level. Equation (2) expresses the CDF calculation.

$$CDF(k) = \sum_{i=0}^k P_k \quad (2)$$

This study converted RGB images into LAB images with three channels: lightness, channel a, and channel b. Subsequently, the histogram equalization was applied to the lightness channel, and then the images were converted back into RGB images. The outcome of the illumination adjustment can be seen in Fig. 5.



Fig. 5. Illumination adjustment

2.2.3. RGB to Gray Conversion

The last preprocessing step converts RGB images to grayscale, combining the R, G, and B channels into a single intensity channel. The conversion formulated in Equation (3) as follows

$$Gray = 0.299 \times R + 0.587 \times G + 0.114 \times B \quad (3)$$

where Gray represents the gray value, R represents the red value, G represents the green value, and B represents the blue value. Fig. 6 shows the outcome of the RGB to gray conversion.



Fig. 6. RGB to gray conversion

2.3. Feature Extraction

2.3.1. LBG-VQ

The Linde Buzo Gray Vector Quantization (LBG-VQ) method designs a codebook representing image texture features [2]. LBG-VQ operates iteratively, starting with an initial codebook of size 1 and using the splitting method to obtain a codebook of sizes 2, 4, 8, and so on. The following steps outline the LBG-VQ codebook design stage [2]:

1. Divide the image into nonoverlapping blocks and convert each block to vectors, thus forming a training vector set.
2. Initialize $i = 1$ (This variable indicates the size of the codebook, i.e., $2^i = 2^1 = 2$).
3. Compute this training vector set's centroid (code vector).

4. Add and subtract constant error i.e., 1, and generate two vectors v_1 and v_2 .
5. Compute the Euclidean distance between all the training vectors belonging to this cluster and the vectors v_1 and v_2 and split the cluster into two.
6. Compute the centroid (code vector) for clusters obtained in the above Step 5.
7. Increment i by one and repeat Step 4 to Step 6 for each code vector.
8. Repeat Step 3 to Step 7 until a codebook of the desired size is obtained.

In this study, LBG-VQ generated a codebook size of 32. LBG-VQ generates a feature vector of size 32×12 for each image and then converts it to 1 dimension, resulting in 384 features per image. Besides codebook generation, the previously generated codebooks were used to obtain the quantized images, which will be combined with MB-LBP later. This technique helps remove unnecessary information and enhance the clarity of features in the data. Fig. 7 shows an example of quantized images.



Fig. 7. Example of a quantized image

Fig. 8 shows the difference of intensity distribution between before and after quantization in a histogram.

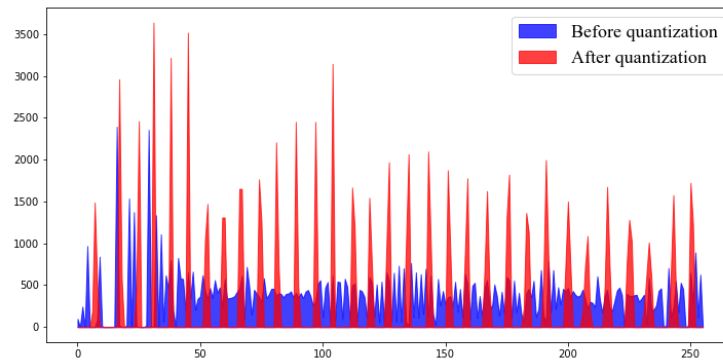
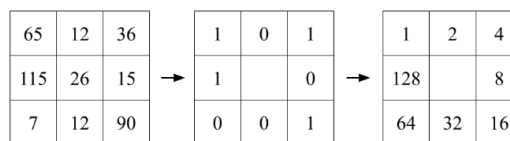


Fig. 8. Histogram Before and after quantization difference

2.3.2. MB-LBP

Multi Block Local Binary Pattern (MB-LBP) leverages the LBP operator to extract features from an image. The LBP operator applied to each pixel and its 8 neighbor pixels within a 3×3 window, utilizes the center pixel as a threshold. For each neighboring pixel, if it is greater than or equal to the threshold, it is set to 1; otherwise, it is set to 0. The LBP operator assigns weights to each pixel, and a new gray value is obtained by summing the neighbor pixels clockwise, starting from the top-left corner [3], [6], [18]. Fig. 9 illustrates the LBP operator.



$$\text{LBP}(10101001) = 1 + 4 + 16 + 128 = 169$$

Fig. 9. Illustration of the LBP operator

MB-LBP divides an image into multiple $3s \times 3t$ windows, subdivided into nine $s \times t$ sub-windows. The average value for each sub-window forms a 3×3 matrix. The LBP operator is then applied to obtain a new gray value [3], [6]. Fig. 10 illustrates the MB-LBP operation, where an image is divided into 9×9 windows, each sub-window being a 3×3 matrix. The average value for each sub-window is calculated, and the LBP operator is subsequently utilized to obtain a new gray value.

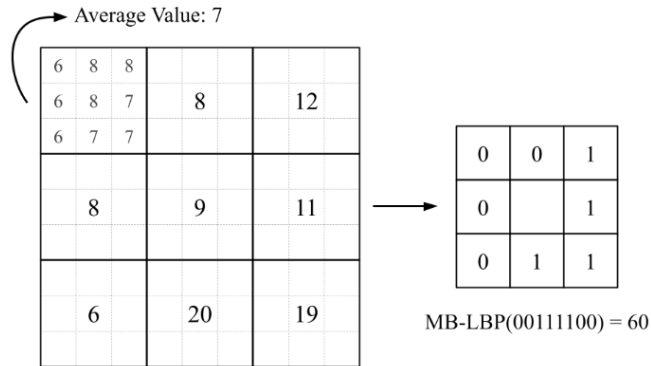


Fig. 10. Illustration of the MB-LBP operation

In this study, MB-LBP was utilized with various window sizes: 3×3 , 6×6 , 9×9 , 12×12 , 15×15 , and 18×18 . When applied independently, MB-LBP extracts features from the preprocessed images. Fig. 11 shows examples of MB-LBP images on the preprocessed images.

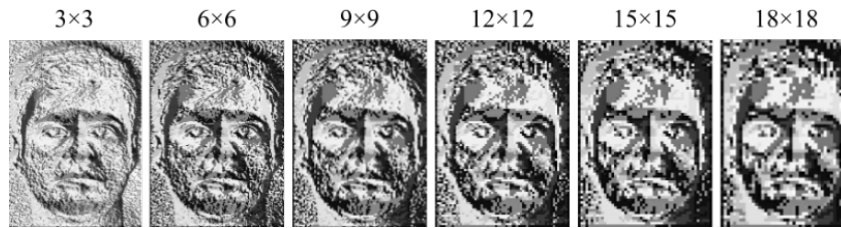


Fig. 11. MB-LBP on the preprocessed image

2.3.3. Combination of LBG-VQ and MB-LBP

Combining both LBG-VQ and MB-LBP aims to enhance feature quality by having LBG-VQ generate quantized images that filter out unnecessary information. MB-LBP then utilizes these refined images to extract texture features, expecting to result in better features. Similar to its independent application, six window sizes were used: 3×3 , 6×6 , 9×9 , 12×12 , 15×15 , and 18×18 . Additionally, MB-LBP effectively captures texture information from these quantized images. Fig. 12 shows examples of MB-LBP on the quantized images.

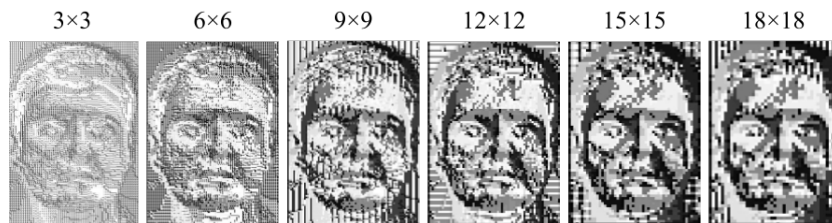


Fig. 12. MB-LBP on the LBG-VQ quantized image

2.4. Modelling

In this stage, classification models were trained using the features obtained in the previous stage. Traditional classifiers were chosen over deep learning techniques to focus on feature extraction methods and facilitate comparisons with prior studies. The classification methods utilized included Naïve Bayes, Support Vector Machine, K Nearest Neighbors, Random Forest, and Logistic Regression. Using the Grid Search technique, the best combination of feature extraction and classification method then selected as the proposed model in this study.

2.4.1. Naïve Bayes

The Naïve Bayes is a classification method leveraging statistical approaches [19] derived from Bayes' Theorem [20], [21]. Gaussian Naïve Bayes, a specific case, applies Gaussian distribution to continuous attributes within given classes [20]–[23]. The calculation is based on Bayes' Theorem equation, which can be seen in Equation (4).

$$p(c|x) = \frac{p(x|c).p(c)}{p(x)} \quad (4)$$

Where c is the class label and x is the attribute; $p(c|x)$ is the probability of class c based on attribute x ; $p(x|c)$ is the probability of an attribute x given class c ; $p(c)$ is the probability of the class label; and $p(x)$ is the probability of the attribute. The formula in the Equation (5) is utilized to calculate the probability of a continuous attribute x which is independent of class c where the μ is the mean and the σ^2 is variance.

$$p(x|c) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (5)$$

There was only one hyperparameter used in this classification method, which describes the portion of the largest variance of all features added. This hyperparameter was set to 1e-9.

2.4.2. Support Vector Machine

Support Vector Machine (SVM) is a supervised learning method commonly utilized in two-class or multiclass classification cases. It separates a set of vectors into two parts based on their class by forming a hyperplane in multidimensional space. If a vector is on one side of the hyperplane, the vector is classified as one class; if in the other side, the vector is classified as another class [19], [24], [25].

SVM establishes the main divider line and two supporting lines. The main divider line separates the two classes, while the support lines are tangent to support vectors. Support vectors are the vectors closest to the main dividing line. The optimal main divider line is obtained by maximizing the distance between these supporting lines [19], [24]–[26].

SVM can also handle non-linear classification using a kernel, which map vectors to a higher dimensional space [26]. In this study, various kernels were used: linear, polynomial, and RBF. Moreover, the parameters for the SVM classifier can be seen in. Two hyperparameters were used in this classifier: the C parameter and gamma. The C parameter, also known as the regularization parameter, was set to 1. The gamma parameter, which is the kernel coefficient for the polynomial and RBF kernels, uses the formula that can be seen in Equation (6).

$$\text{gamma} = \frac{1}{\text{number of features} \times \text{features variance}} \quad (6)$$

2.4.3. K Nearest Neighbors

K Nearest Neighbors (KNN) is a supervised learning algorithm that labels input data before training. It classifies data based on proximity to its neighbors, using Euclidian Distance [27]–[30]. Equation (7) shows the Euclidian Distance formula.

$$\text{dist}(X, Y) = \sqrt{\sum_{i=1}^n (X_i - Y_i)^2} \quad (7)$$

The parameter K influences the prediction results, representing the number of neighbors considered to determine the class. KNN predicts the class of data by selecting the majority class among its K neighbors [27], [28], [30]. Fig. 13 illustrates the KNN classification: with $K = 1$, the data is predicted to belong to class B; with $K = 3$, the prediction assigns the data to class A.

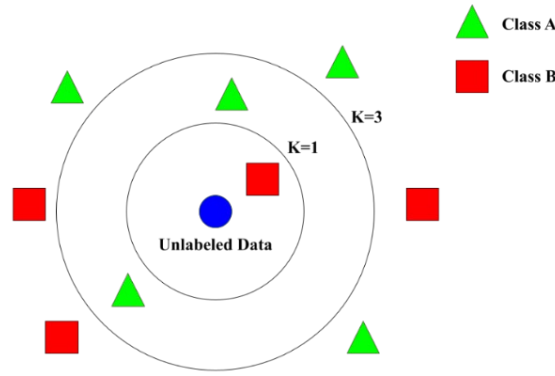


Fig. 13. Illustration of the K Nearest Neighbors

In this study, the main hyperparameter K , was set to 5. This means the classifier uses five nearest neighbors to determine the class of a given data point.

2.4.4. Random Forest

Random Forest (RF) is an ensemble learning method for classification tasks [31]–[34]. It constructs an ensemble of multiple decision trees, a classification method with nodes representing the prediction result [19]. These decision trees are then merged and leveraged to predict data classes [33].

Initially, the dataset is split into training data and testing data. The next step involves sampling with replacement in the training data, generating several bootstrap samples. Each bootstrap sample serves as training material for an individual Decision Tree model. Subsequently, the predictions from these Decision Tree models are aggregated, forming a Random Forest classifier. The prediction result for unlabeled data is determined by taking the majority of predictions from the combined Decision Tree models [31], [32].

In this study, several hyperparameters were used. The number of trees was set to 100, meaning a random forest creates 100 trees. The criterion was set to Gini, indicating that Gini impurity was used to measure the quality of a decision tree split. The minimum samples split parameter, which specifies the minimum number of samples required to split an internal node, was set to 2. The minimum sample leaf parameter, which specifies the minimum number of samples a node must hold after splitting, was set to 1. The maximum features parameter, which determines the number of features to consider for the best split, was set to the square root of the number of features.

2.4.5. Logistic Regression

Logistic Regression (LGR), also known as the logistic model or the logit model, is a method often utilized in binary classification cases [35]–[38]. The general form of logistic regression for binary classification can be seen in Equation (8) and Equation (9) [36], [39].

$$p(x) = \frac{e^{(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n)}}{1 + e^{(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n)}} \quad (8)$$

$$y = \ln\left(\frac{p(x)}{1-p(x)}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n \quad (9)$$

where $p(x)$ is the regression probability value and has a value range from 0 to 1; β_0 is the constant, some define it as the intercept; $\beta_1 + \beta_2 + \dots + \beta_n$ are the coefficients to be estimated; $x_1 + x_2 + \dots + x_n$ are the independent variables, or some might call as the predictors; and y is the dependent variable.

Similar to other classifiers, Logistic Regression uses several hyperparameters. The l2 regularization was used as the penalty for this classifier to reduce the values of the coefficients nearly to zero. The C parameter, or regularization strength, was set to 1. The solver parameter, which sets the algorithm used to optimize the model, was set to Limited-memory Broyden–Fletcher–Goldfarb–Shanno algorithm (L-BFGS).

2.5. Evaluation

K-Fold Cross Validation divides a dataset into K folds, with K-1 folds used as training data and the remaining fold as testing data. This method performs K iterations, evaluating the model with different combinations of training and testing data in each iteration [29], [40]. In this study, K was set to 10, resulting in 10 iterations of modeling and evaluation, each using different data for training and testing. Fig. 14 illustrates the Stratified K Fold Cross Validation process.

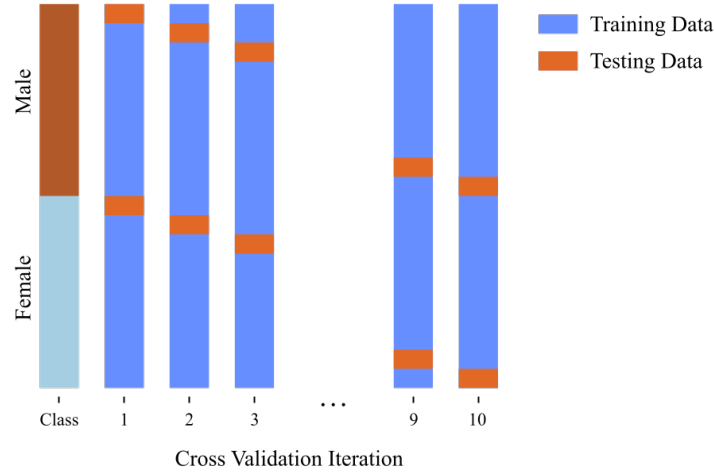


Fig. 14. Stratified K-Fold Cross Validation

The evaluation metric chosen was the Macro F1 Score, calculated by averaging F1 Scores for the male and female classes. The F1 Score itself is the harmonic average of Precision and Recall. Equation (10), (11), (12), and (13) represent the formulas for Precision, Recall, F1 Score, and Macro F1 Score, respectively [41].

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (10)$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (11)$$

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (12)$$

$$Macro\ F1\ Score = \frac{\sum_{i=1}^n F1\ Score_i}{n} \quad (13)$$

3. Results and Discussion

This section discusses the findings of this study. The detailed stages to obtain the results can be found in Section 2. Thirteen different features were utilized in this study, including LBG-VQ codebooks, MB-LBP with six different parameters, and MB-LBP on quantized images with six different parameters. Each feature underwent a split into training and testing data using K-Fold Cross Validation, where K was set to 10. This configuration resulted in 10 iterations, with each iteration containing 9 folds of training data and 1 fold of testing data.

The training data were then utilized to train machine learning models with various classification methods: Naïve Bayes, SVM with linear kernel, SVM with polynomial kernel, SVM with RBF kernel, KNN, RF, and LGR. Consequently, a total of 91 performance classification results were generated by combining different feature extraction methods with various classification methods.

The K-Fold Cross Validation implemented in this study was stratified, ensuring an even distribution of class labels between training and testing data. With 10 iterations, 10 classification performance values were obtained, each calculated using the Macro F1 Score as the evaluation metric. Subsequently, the

average of these values was computed to represent the overall performance of the machine learning models based on the feature extraction and classification methods employed. Fig. 15 shows the overall model classification performance results.

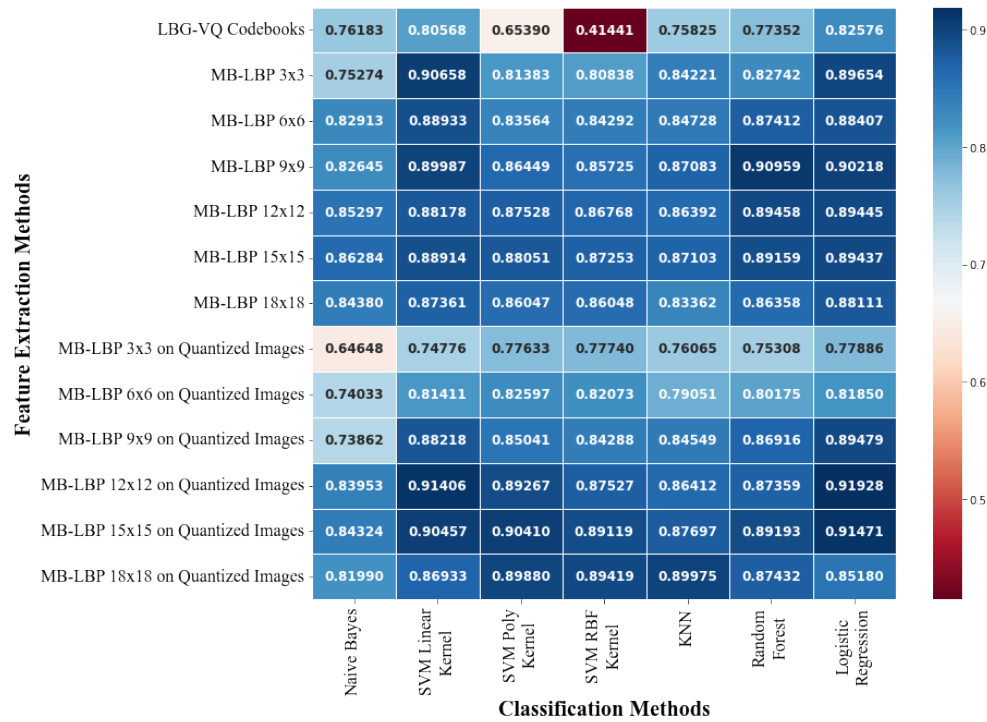


Fig. 15. Classification Performance Results

There are variations in classification performance when using MB-LBP compared to LBG-VQ. The extent of improvement or reduction depends on the chosen MB-LBP parameters. In general, MB-LBP tends to yield higher accuracy than LBG-VQ. Furthermore, applying MB-LBP to LBG-VQ features (quantized images) also tends to result in better classification results than using LBG-VQ independently.

However, there were cases where LBG-VQ codebooks outperform MB-LBP, such as with Naïve Bayes, SVM with Linear Kernel, Random Forest, and Logistic Regression, where LBG-VQ codebooks outperform MB-LBP 3x3 on quantized images. On the other hand, other classifiers show opposite results, even obtaining the lowest performance score.

The highest level of performance, reaching 91.928%, was achieved when utilizing Logistic Regression as the classification method. This was achieved with MB-LBP with a window size of 12x12 applied to LBG-VQ quantized images. This combination was then selected as the proposed model for gender classification in this study. The confusion matrix of the best-performing model is shown in Fig. 16.



Fig. 16. Confusion Matrix of the Best Performing Model

There are many factors that contribute to why Logistic Regression outperforms other classifiers, such as proper feature selection, with MB-LBP 12x12 on Quantized Images being quite suitable in this case. As shown in Fig. 16, the averaged 10-fold cross-validation confusion matrix demonstrates the model's overall performance. The higher numbers on the diagonal (True Female, Predicted Female and True Male, Predicted Male) indicate that the model is making correct predictions more frequently, thus reflecting its reliability. Moreover, Table 1 and Table 2 show a statistical approach for further analysis.

Table 1. Statistical Performance of Feature Extraction Methods

Feature Extraction Methods	Mean	Standard Deviation
LBG-VQ Codebooks	0.713336	0.142622
MB-LBP 3x3	0.835386	0.053139
MB-LBP 6x6	0.857499	0.024478
MB-LBP 9x9	0.875809	0.029858
MB-LBP 12x12	0.875809	0.015625
MB-LBP 15x15	0.880287	0.011936
MB-LBP 18x18	0.859524	0.016334
MB-LBP 3x3 on Quantized Images	0.748651	0.046741
MB-LBP 6x6 on Quantized Images	0.801700	0.029662
MB-LBP 9x9 on Quantized Images	0.846219	0.051292
MB-LBP 12x12 on Quantized Images	0.882646	0.028190
MB-LBP 15x15 on Quantized Images	0.889530	0.023740
MB-LBP 18x18 on Quantized Images	0.872584	0.029193

When LBG-VQ Codebooks were utilized, it resulted in the lowest mean value of 71.33% and the highest standard deviation of 14.26%, meaning LBG-VQ Codebooks were unreliable due to the wide spread of performance scores. In contrast, many features were reliable, with high mean values and low standard deviations. For example, MB-LBP 15x15 on Quantized Images has the highest mean value of 88.95% and a low standard deviation of 2.37%. Similarly, MB-LBP 15x15 has a high mean value of 88.02% and the lowest standard deviation of 1.19%. The proposed model in this study, which utilizes MB-LBP 12x12 applied to quantized images, also shows good statistics, with a high mean value of 88.26% and a low standard deviation of 2.81%.

Table 2. Statistical Performance of Classification Methods

Classification Methods	Mean	Standard Deviation
Naïve Bayes	0.796758	0.063574
SVM Linear Kernel	0.867538	0.048761
SVM Poly Kernel	0.840954	0.067038
SVM RBF Kernel	0.817332	0.125531
KNN	0.840356	0.044309
Random Forest	0.853710	0.049337
Logistic Regression	0.873571	0.042206

From Table 2, it can be observed that the proposed model, which utilizes Logistic Regression as a classifier, has the best statistics, with the highest mean value of 87.35% and the lowest standard deviation of 4.2%. This makes Logistic Regression a reliable and optimal choice for classifiers. On the other hand, Naïve Bayes got the lowest mean value of 79.67%, and SVM with RBF kernel got the highest standard deviation of 12.55%.

This study utilizes the FEI facial images dataset, which lacks diversity of ethnicity, age groups, and real-world variations. While the traditional classifier methods utilized in this study were effective, they may not capture complex facial features or handle extensive datasets as efficiently as deep learning models. Therefore, future studies may adopt a similar approach to this study, with expanded datasets that include more diverse facial images, utilizing deep learning techniques, and evaluating real-time efficiency under various hardware constraints. Furthermore, investigating the combination of LBG-VQ and MB-LBP with other hybrid feature extraction methods could be a valuable direction for future research.

4. Conclusion

Based on the conducted study, it can be concluded that gender classification performance on the FEI facial image dataset has improved compared to previous studies. The utilization of the MB-LBP method demonstrated superior performance compared to the LBG-VQ method. Furthermore, applying MB-LBP to LBG-VQ features (quantized images) achieved even better results. The highest-performing machine learning model, reaching 91.928% accuracy, utilized the Logistic Regression classification method and MB-LBP 12x12 feature extraction method on LBG-VQ quantized images. This model was obtained through a three-step preprocessing stage: noise removal, illumination adjustment, and RGB to gray conversion.

Acknowledgment

The deepest gratitude is expressed to all those who have contributed to completing this study entitled "Gender classification performance optimization based on facial images using LBG-VQ and MB-LBP feature extraction methods." Moreover, thanks to the Faculty of Computer Science, University of Jember, for supporting this study. Hopefully, this study can advance technology, especially in computer vision and machine learning.

Declarations

Author contribution. All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper

Funding statement. This study has not received any funding.

Conflict of interest. The authors declare no conflict of interest.

Additional information. No additional information is available for this paper.

References

- [1] A. Swaminathan, M. Chaba, D. K. Sharma, and Y. Chaba, "Gender Classification using Facial Embeddings: A Novel Approach," *Procedia Comput. Sci.*, vol. 167, pp. 2634–2642, 2020, doi: [10.1016/j.procs.2020.03.342](https://doi.org/10.1016/j.procs.2020.03.342).
- [2] S. R. Shinde and S. Thepade, "Gender Classification from Face Images Using LBG Vector Quantization with Data Mining Algorithms," in *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBE)*, Aug. 2018, pp. 1–5, doi: [10.1109/ICCUBE.2018.8697784](https://doi.org/10.1109/ICCUBE.2018.8697784).
- [3] L. Tianyu, L. Fei, and W. Rui, "Human face gender identification system based on MB-LBP," in *2018 Chinese Control And Decision Conference (CCDC)*, Jun. 2018, pp. 1721–1725, doi: [10.1109/CCDC.2018.8407405](https://doi.org/10.1109/CCDC.2018.8407405).
- [4] A. Bakshi *et al.*, "Performance Augmentation of Cuckoo Search Optimization Technique Using Vector Quantization in Image Compression," *Mathematics*, vol. 11, no. 10, p. 2364, May 2023, doi: [10.3390/math1102364](https://doi.org/10.3390/math1102364).
- [5] M. Bilal, Z. Ullah, O. Mujahid, and T. Fouzder, "Fast Linde–Buzo–Gray (FLBG) Algorithm for Image Compression through Rescaling Using Bilinear Interpolation," *J. Imaging*, vol. 10, no. 5, p. 124, May 2024, doi: [10.3390/jimaging10050124](https://doi.org/10.3390/jimaging10050124).
- [6] S. Alizadeh, H. B. Jond, V. V. Nabyeyev, and C. Kose, "Automatic Retrieval of Shoeprints Using Modified Multi-Block Local Binary Pattern," *Symmetry (Basel)*, vol. 13, no. 2, p. 296, Feb. 2021, doi: [10.3390/sym13020296](https://doi.org/10.3390/sym13020296).
- [7] M. S. Karis, N. R. A. Razif, N. M. Ali, M. A. Rosli, M. S. M. Aras, and M. M. Ghazaly, "Local Binary Pattern (LBP) with application to variant object detection: A survey and method," in *2016 IEEE 12th International Colloquium on Signal Processing & Its Applications (CSPA)*, Mar. 2016, pp. 221–226, doi: [10.1109/CSPA.2016.7515835](https://doi.org/10.1109/CSPA.2016.7515835).
- [8] Wu Jiang, Fei Ding, and Qiao-Liang Xiang, "An affinity propagation based method for vector quantization codebook design," in *2008 19th International Conference on Pattern Recognition*, Dec. 2008, pp. 1–4, doi: [10.1109/ICPR.2008.4761392](https://doi.org/10.1109/ICPR.2008.4761392).

- [9] I. A. Siradjuddin, H. A. Achmani, and A. Muntasa, "Adaptive Boosting Classifier for Pedestrian Attributes Identification with Color and Texture Features," in *2018 International Conference on Computer Engineering, Network and Intelligent Multimedia (CENIM)*, Nov. 2018, pp. 275–279, doi: [10.1109/CENIM.2018.8710845](https://doi.org/10.1109/CENIM.2018.8710845).
- [10] "FEI Face Database," FEI. <https://fei.edu.br/~cet/facedatabase.html>.
- [11] M. G. Tatuin, Y. P. . Kelen, and S. S. Manek, "Pengaruh Ukuran Jendela Ketetangaan (Window) Terhadap Hasil Redukasi Noise pada Metode Median Filter dan Gaussian Filter," *J. Krisnadana*, vol. 3, no. 3, pp. 142–154, May 2024, doi: [10.58982/krisnadana.v3i3.601](https://doi.org/10.58982/krisnadana.v3i3.601).
- [12] H. H. Draz, N. E. Elashker, and M. M. A. Mahmoud, "Optimized Algorithms and Hardware Implementation of Median Filter for Image Processing," *Circuits, Syst. Signal Process.*, vol. 42, no. 9, pp. 5545–5558, Sep. 2023, doi: [10.1007/s00034-023-02370-x](https://doi.org/10.1007/s00034-023-02370-x).
- [13] K. Mishiba, "Fast Guided Median Filter," *IEEE Trans. Image Process.*, vol. 32, pp. 737–749, 2023, doi: [10.1109/TIP.2022.3232916](https://doi.org/10.1109/TIP.2022.3232916).
- [14] B. Bataineh, "Image contrast enhancement for preserving entropy and image visual features," *Int. J. Adv. Intell. Informatics*, vol. 9, no. 2, p. 161, Jul. 2023, doi: [10.26555/ijain.v9i2.907](https://doi.org/10.26555/ijain.v9i2.907).
- [15] P. Härtinger and C. Steger, "Adaptive histogram equalization in constant time," *J. Real-Time Image Process.*, vol. 21, no. 3, p. 93, Jun. 2024, doi: [10.1007/s11554-024-01465-1](https://doi.org/10.1007/s11554-024-01465-1).
- [16] R. M. Dyke and K. Hormann, "Histogram equalization using a selective filter," *Vis. Comput.*, vol. 39, no. 12, pp. 6221–6235, Dec. 2023, doi: [10.1007/s00371-022-02723-8](https://doi.org/10.1007/s00371-022-02723-8).
- [17] D. Xiang, H. Wang, D. He, and C. Zhai, "Research on Histogram Equalization Algorithm Based on Optimized Adaptive Quadruple Segmentation and Cropping of Underwater Image (AQSCHE)," *IEEE Access*, vol. 11, pp. 69356–69365, 2023, doi: [10.1109/ACCESS.2023.3290201](https://doi.org/10.1109/ACCESS.2023.3290201).
- [18] U. Lakshmi Sowjanya and R. Krithiga, "Decoding Student Emotions: An Advanced CNN Approach for Behavior Analysis Application Using Uniform Local Binary Pattern," *IEEE Access*, vol. 12, pp. 106273–106284, 2024, doi: [10.1109/ACCESS.2024.3436531](https://doi.org/10.1109/ACCESS.2024.3436531).
- [19] Dewi Widyawati and Amaliah Faradibah, "Comparison Analysis of Classification Model Performance in Lung Cancer Prediction Using Decision Tree, Naive Bayes, and Support Vector Machine," *Indones. J. Data Sci.*, vol. 4, no. 2, pp. 80–89, Jul. 2023, doi: [10.56705/ijodas.v4i2.76](https://doi.org/10.56705/ijodas.v4i2.76).
- [20] S. Naiem, A. E. Khedr, A. M. Idrees, and M. I. Marie, "Enhancing the Efficiency of Gaussian Naïve Bayes Machine Learning Classifier in the Detection of DDOS in Cloud Computing," *IEEE Access*, vol. 11, pp. 124597–124608, 2023, doi: [10.1109/ACCESS.2023.3328951](https://doi.org/10.1109/ACCESS.2023.3328951).
- [21] I. Akil and I. Chaidir, "Classification of Heart Disease Diagnoses Using Gaussian Naïve Bayes," *Komputasi J. Ilm. Ilmu Komput. dan Mat.*, vol. 21, no. 2, pp. 31–36, Aug. 2024, doi: [10.33751/komputasi.v21i2.10114](https://doi.org/10.33751/komputasi.v21i2.10114).
- [22] C. D. Suhendra, E. Najwaini, E. Maria, and E. Faizal, "A Machine Learning Perspective on Daisy and Dandelion Classification: Gaussian Naive Bayes with Sobel," *Indones. J. Data Sci.*, vol. 4, no. 3, pp. 151–159, Dec. 2023, doi: [10.56705/ijodas.v4i3.112](https://doi.org/10.56705/ijodas.v4i3.112).
- [23] M. Hafidz Ariansyah, Esmi Nur Fitri, and Sri Winarno, "Improving Performance Of Students' Grade Classification Model Uses Naïve Bayes Gaussian Tuning Model And Feature Selection," *J. Tek. Inform.*, vol. 4, no. 3, pp. 493–501, Jun. 2023, doi: [10.52436/1.jutif.2023.4.3.737](https://doi.org/10.52436/1.jutif.2023.4.3.737).
- [24] H. K. Easa, A. A. Saber, N. K. Hamid, and H. A. Saber, "Machine learning based approach for detection of fake banknotes using support vector machine," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 31, no. 2, p. 1016, Aug. 2023, doi: [10.11591/ijeecs.v31.i2.pp1016-1022](https://doi.org/10.11591/ijeecs.v31.i2.pp1016-1022).
- [25] M. D. Purbolaksono, D. T. B. Pratama, and F. Hamzah, "Perbandingan Gini Index dan Chi Square pada Sentimen Analsis Ulasan Film menggunakan Support Vector Machine Classifier," *J. Edukasi dan Penelit. Inform.*, vol. 9, no. 3, p. 528, Dec. 2023, doi: [10.26418/jp.v9i3.68845](https://doi.org/10.26418/jp.v9i3.68845).
- [26] Y. Yan, X. Le, T. Yang, and H. Yu, "Interpretable PCA and SVM-Based Leak Detection Algorithm for Identifying Water Leakage Using SAR-Derived Moisture Content and InSAR Closure Phase," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 17, pp. 15136–15147, 2024, doi: [10.1109/JSTARS.2024.3443127](https://doi.org/10.1109/JSTARS.2024.3443127).

- [27] D. T. Wilujeng, M. Fatekurohman, and I. M. Tirta, "Analisis Risiko Kredit Perbankan Menggunakan Algoritma K-Nearest Neighbor dan Nearest Weighted K-Nearest Neighbor," *Indones. J. Appl. Stat.*, vol. 5, no. 2, p. 142, Oct. 2023, doi: [10.13057/ijas.v5i2.58426](https://doi.org/10.13057/ijas.v5i2.58426).
- [28] U. Suriani and T. B. Kurniawan, "Comparing the Prediction of Numeric Patterns on Form C1 Using the K-Nearest Neighbors (K-NN) Method and a Combination of K-Nearest Neighbors (K-NN) with Connected Component Labeling (CCL)," *J. Inf. Syst. Informatics*, vol. 5, no. 4, pp. 1569–1580, Dec. 2023, doi: [10.51519/journalisi.v5i4.592](https://doi.org/10.51519/journalisi.v5i4.592).
- [29] I. G. Iwan Sudipa, R. A. Azdy, I. Arfiani, N. M. Setiohardjo, and Sumiyatun, "Leveraging K-Nearest Neighbors for Enhanced Fruit Classification and Quality Assessment," *Indones. J. Data Sci.*, vol. 5, no. 1, pp. 30–36, Mar. 2024, doi: [10.56705/ijodas.v5i1.125](https://doi.org/10.56705/ijodas.v5i1.125).
- [30] F. T. Admojo, M. L. Radhitya, H. Zein, and A. Naswin, "Classification of Mushroom Edibility Using K-Nearest Neighbors: A Machine Learning Approach," *Indones. J. Data Sci.*, vol. 5, no. 3, pp. 243–250, Dec. 2024, doi: [10.56705/ijodas.v5i3.199](https://doi.org/10.56705/ijodas.v5i3.199).
- [31] I. Ilham, "Predicting Plant Growth Stages Using Random Forest Classifier: A Machine Learning Approach," *Indones. J. Data Sci.*, vol. 5, no. 2, Jul. 2024, doi: [10.56705/ijodas.v5i2.167](https://doi.org/10.56705/ijodas.v5i2.167).
- [32] M. Mafarja *et al.*, "Classification framework for faulty-software using enhanced exploratory whale optimizer-based feature selection scheme and random forest ensemble learning," *Appl. Intell.*, vol. 53, no. 15, pp. 18715–18757, Aug. 2023, doi: [10.1007/S10489-022-04427-X/TABLES/12](https://doi.org/10.1007/S10489-022-04427-X/TABLES/12).
- [33] T. E. Tarigan, E. Susanti, M. I. Siami, I. Arfiani, A. A. Jiwa Permana, and I. M. Sunia Raharja, "Performance Metrics of AdaBoost and Random Forest in Multi-Class Eye Disease Identification: An Imbalanced Dataset Approach," *Int. J. Artif. Intell. Med. Issues*, vol. 1, no. 2, pp. 84–94, Nov. 2023, doi: [10.56705/ijaimi.v1i2.98](https://doi.org/10.56705/ijaimi.v1i2.98).
- [34] Siti Khomsah and Edi Faizal, "Effectiveness Evaluation of the RandomForest Algorithm in Classifying CancerLips Data," *Int. J. Artif. Intell. Med. Issues*, vol. 1, no. 1, pp. 10–17, May 2023, doi: [10.56705/ijaimi.v1i1.84](https://doi.org/10.56705/ijaimi.v1i1.84).
- [35] A. Hadianfar *et al.*, "Predictors of in-hospital mortality among patients with symptoms of stroke, Mashhad, Iran: an application of auto-logistic regression model," *Arch. Public Heal.*, vol. 81, no. 1, p. 73, Apr. 2023, doi: [10.1186/s13690-023-01084-5](https://doi.org/10.1186/s13690-023-01084-5).
- [36] G. Bordeaux and A. Couto, "A binary logistic regression model to identify key aspects that enhance global hub airports status," *J. Air Transp. Res. Soc.*, vol. 3, p. 100035, Dec. 2024, doi: [10.1016/j.jatrs.2024.100035](https://doi.org/10.1016/j.jatrs.2024.100035).
- [37] D. M. A. S. Elkahwagy, C. J. Kiriacos, and M. Mansour, "Logistic regression and other statistical tools in diagnostic biomarker studies," *Clin. Transl. Oncol.*, vol. 26, no. 9, pp. 2172–2180, Mar. 2024, doi: [10.1007/s12094-024-03413-8](https://doi.org/10.1007/s12094-024-03413-8).
- [38] S. Park, E. Ceulemans, and K. Van Deun, "Logistic regression with sparse common and distinctive covariates," *Behav. Res. Methods*, vol. 55, no. 8, pp. 4143–4174, Feb. 2023, doi: [10.3758/s13428-022-02011-2](https://doi.org/10.3758/s13428-022-02011-2).
- [39] I. Alwiah, U. Zaky, and A. W. Murdiyanto, "Assessing the Predictive Power of Logistic Regression on Liver Disease Prevalence in the Indian Context," *Indones. J. Data Sci.*, vol. 5, no. 1, pp. 1–7, Mar. 2024, doi: [10.56705/ijodas.v5i1.121](https://doi.org/10.56705/ijodas.v5i1.121).
- [40] T. Ait tchakoucht, B. Elkari, Y. Chaibi, and T. Kousksou, "Random forest with feature selection and K-fold cross validation for predicting the electrical and thermal efficiencies of air based photovoltaic-thermal systems," *Energy Reports*, vol. 12, pp. 988–999, Dec. 2024, doi: [10.1016/j.egyr.2024.07.002](https://doi.org/10.1016/j.egyr.2024.07.002).
- [41] K. Takahashi, K. Yamamoto, A. Kuchiba, and T. Koyama, "Confidence interval for micro-averaged F1 and macro-averaged F1 scores," *Appl. Intell.*, vol. 52, no. 5, pp. 4961–4972, Mar. 2022, doi: [10.1007/s10489-021-02635-5](https://doi.org/10.1007/s10489-021-02635-5).