An Efficient Weighted Majority Voting Ensemble Machine Learning Classifier Framework for Image Segmentation

Zahra Faska
Laboratory of Applied Sciences and Emerging Technologies, ENSA, Sidi Mohamed Ben Abdellah University, Fez, Morocco
zahra.faska@usmba.ac.ma

Lahbib Khrissi
Laboratory of Applied Sciences and Emerging Technologies, ENSA, Sidi Mohamed Ben Abdellah University, Fez, Morocco
lahbib.khrissi@usmba.ac.ma

Imadeddine Mountasser
Laboratory of Applied Sciences and Emerging Technologies, ENSA, Sidi Mohamed Ben Abdellah University, Fez, Morocco
imadeddine.mountasser@usmba.ac.ma

Khalid Haddouch
Laboratory of Applied Sciences and Emerging Technologies, ENSA, Sidi Mohamed Ben Abdellah University, Fez, Morocco
khalid.haddouch@usmba.ac.ma

Nabil El Akkad
Laboratory of Applied Sciences and Emerging Technologies, ENSA, Sidi Mohamed Ben Abdellah University, Fez, Morocco
nabil.elakkad@usmba.ac.ma

Walid El-Shafai
Automated Systems and Computing Lab, Prince Sultan University, Riyadh, Saudi Arabia | Department of Electronics and Electrical Communications Engineering, Faculty of Electronic Engineering, Menoufia University, Menoufia, Egypt
welshafai@psu.edu.sa

Abrar Fallatah
College of Computer and Information Sciences, Prince Sultan University, Riyadh, Saudi Arabia | Automated Systems and Computing Lab, Prince Sultan University, Riyadh, Saudi Arabia
afallatah@psu.edu.sa (corresponding author)

Ahmad Taher Azar
College of Computer and Information Sciences, Prince Sultan University, Riyadh, Saudi Arabia | Automated Systems and Computing Lab, Prince Sultan University, Riyadh, Saudi Arabia
aazar@psu.edu.sa

Saim Ahmed
Automated Systems and Computing Lab, Prince Sultan University, Riyadh, Saudi Arabia
sahmed@psu.edu.sa

Received: 30 November 2025 | Revised: 2 February 2026 | Accepted: 13 February 2026

 DOI: https://doi.org/10.48084/etasr.16623

ABSTRACT

Ensemble learning is an effective approach to improving the robustness and accuracy of image segmentation by combining multiple classifiers. This study presents an efficient Weighted Majority Voting Ensemble (WMVE) framework integrating five Machine Learning (ML)-based segmentation models: Random Forest (RF), Naïve Bayes (NB), eXtreme Gradient Boosting (XGB), K-Nearest Neighbors (KNN), and Multilayer Perceptron (MLP). The standard preprocessing techniques include Gaussian smoothing and median filtering to ensure reliable feature extraction and high-quality segmentation. The proposed WMVE allows assigning specific weights to each classifier according to its validation accuracy result for adaptive informed decision fusion, where performance evaluation uses region-based segmentation metrics, including Segmentation Covering (SC), Probabilistic Rand Index (PRI), Variation of Information (VoI), Global Consistency Error (GCE), and Boundary Displacement Error (BDE). Experimental results indicate that the proposed ensemble is better than single classifiers and nearly as good as existing ensemble and deep clustering approaches on several datasets. Therefore, the WMVE framework can be considered a strong approach to attain high performance in image segmentation, since experimental results also show near-optimal performance with existing state-of-the-art methods.

Keywords-image segmentation; ensemble learning; random forest; KNN; Bayesian network; XGB; MLP; voting; weighted majority voting

I. INTRODUCTION

Image segmentation, a crucial task in computer vision, plays a significant role in diverse applications such as intelligent medical systems, autonomous vehicles, image retrieval, video surveillance, augmented reality, and industrial inspection. It partitions an image into distinct regions or contours based on pixel intensity, color, and texture. Unlike image classification, which assigns labels to objects, segmentation aims to delineate both known and unknown objects into uniform, homogeneous regions, a task that remains highly challenging. Numerous segmentation algorithms have been developed [1-4], with traditional approaches frequently employing clustering techniques, particularly those based on the Markov model.

Accurate image segmentation relies heavily on digital image processing and mathematical modeling but is prone to noise and requires substantial human-computer interaction, despite relying heavily on digital image processing and mathematical modeling [5]. Since individual classification models exhibit varying strengths and weaknesses depending on the dataset, prediction accuracy can fluctuate. To address these limitations, researchers have focused on optimizing classification performance and reducing prediction errors by leveraging established algorithms.

Ensemble learning is an effective solution used to enhance prediction accuracy and model robustness [6]. By integrating multiple models and combining their outputs through averaging or voting mechanisms, ensemble methods exploit the complementary strengths of individual classifiers. This multi-model strategy significantly improves the precision, generalization, and efficiency of classification systems compared to single-model approaches. Compared to a single model, ensemble learning combines multiple models to achieve higher accuracy and generalization. Its main techniques, bagging, boosting, stacking, and voting, differ in structure and objective: bagging reduces variance through bootstrap sampling and independent training; boosting strengthens weak learners by weighting misclassified instances, such as in AdaBoost; stacking employs a meta-learner to integrate heterogeneous base classifiers; and voting aggregates model outputs through majority or weighted schemes [7].

Despite their advantages, ensemble methods present challenges, including high computational demands, large training data requirements, potential overfitting when models are correlated, architectural complexity, and limited transferability across domains. Nevertheless, the benefits generally outweigh these limitations. Ensemble models enhance accuracy, robustness, and generalization by balancing bias and variance, effectively handling complex or noisy data. By merging complementary model strengths, ensemble learning reduces prediction errors and improves reliability through collective decision-making, yielding more stable and dependable results.

The main focus of this research centers around the voting process employed by ensemble members, specifically multiple classifiers, to determine a conclusive prediction for a given ensemble. To enhance accuracy, the widely utilized technique of voting is employed in both human and artificial intelligence. This involves merging the predictions from each classifier and selecting the most closely aligned results to determine the optimal prediction. The underlying theory behind voting is straightforward: assuming that most individual models are relatively accurate, with more correct predictions than incorrect ones, and have a certain level of independence, the majority of predictions will be correct in most situations, while only a minority will be incorrect. Therefore, by utilizing voting, the present study increases the probability of obtaining correct

predictions in a greater number of situations compared to relying solely on individual classifiers.

Methods based on voting in ensemble learning utilize various learning algorithms, thereby enhancing the robustness of the classification model. In contrast to unweighted (majority) voting ensemble methods, weighted voting-based ensemble techniques offer a more nuanced and adaptable approach for predicting the true output classes. However, in weighted voting, the classifiers have differing levels of impact on the ultimate prediction. Each classifier is assigned a coefficient or weight, typically based on its accuracy in classifying a validation set. The final determination is made by adding up the weighted votes and selecting the class with the highest cumulative score.

Despite the fact that weighted voting ensembles have already been thoroughly studied in the literature, many existing approaches depend on some complex weight optimization strategies or an iterative learning procedure, increasing computational cost and making it hard for practical application. Weighted Majority Voting Ensemble (WMVE) has been proposed as a Cross-Validation (CV) performance lightweight and efficient framework directly assigning weights to classifiers without any extra optimization stage. More specifically, another important difference between generic ensemble methods focused mainly on classification accuracy, and WMVE is that the latter is formulated particularly for image segmentation; hence, stressing region-based performance criteria during model selection. The proposed method combines different types of Machine Learning (ML) classifiers possessing complementary learning behaviors into one composite adaptive voting mechanism, thereby improving both robustness and generalization at low computational complexity.

The primary objective of this paper is to present a fresh and innovative technique that involves the utilization of an ensemble method. This approach combines numerous diverse ML-based segmentation models. Through experimentation, it has been demonstrated that this novel approach leads to enhanced performance compared to using individual ML-based segmentation models.

The main outcomes of this research can be summarized as:

- A WMVE method was proposed as an effective strategy for integrating multiple, diverse ML-based segmentation models. The ensemble dynamically adjusts the influence of individual models by reducing the impact of inaccurate predictors and amplifying that of accurate ones.
- The proposed framework consists of three distinct phases: (1) training classifiers using the training dataset; (2) determining classifier weights through validation, where the weight of each accurate classifier is increased relative to the ratio of misclassified instances; and (3) combining classifier outputs based on their assigned weights to produce the final segmentation.
- Extensive experiments on multiple datasets demonstrated that the WMVE approach consistently achieved superior classification accuracy compared to individual baseline algorithms.
- The performance of the ML-based segmentation models was rigorously evaluated using quantitative metrics, such as Segmentation Covering (SC), Probabilistic Rand Index (PRI), Variation of Information (VoI), Boundary Displacement Error (BDE), and Global Consistency Error (GCE), alongside additional evaluation measures.

Thus, the main contributions of this work include:

- A comparative evaluation of five state-of-the-art portrait segmentation models, Random Forest (RF), Naïve Bayes (NB), eXtreme Gradient Boosting (XGB), K-Nearest Neighbours (KNN), and Multilayer Perceptron (MLP), selected for their proven effectiveness and widespread adoption in image segmentation tasks.
- The successful development of a set of individual portrait segmentation models integrated through a WMV method.
- The execution of a comprehensive quantitative assessment to validate the segmentation performance and overall robustness of the proposed ensemble framework.

II. RELATED WORKS

A. Ensemble Learning Techniques

In ML, ensemble methods have been extensively adopted to build high-performing models by combining multiple classifiers to enhance accuracy and effectiveness in classification tasks. There has been renewed interest in ensemble classification models that integrate different learners to achieve superior performance. By leveraging the collective predictions of multiple classifiers, ensemble learning generates more accurate and reliable outcomes for new instances. This trend reflects the increasing recognition of ensemble techniques as powerful tools for improving model generalization and robustness. Typically, the process involves exploring various classifiers and selecting an optimal subset to form the ensemble. The integration of these classifiers and their predictions results in significantly improved performance compared to using any single model.

Ensemble approaches yield stronger generalization than individual learners. One of their primary advantages lies in transforming weak learners into highly accurate models through aggregation. Rather than depending on a single algorithm to build a predictive model, ensemble methods combine multiple classifiers, strong or weak, referred to as base learners [8]. The effectiveness of an ensemble largely depends on the appropriate selection of base learners and the fusion strategy employed to integrate them into the final hypothesis. For optimal performance, each learner should generate a diverse set of predictions, producing unique models. By analyzing relationships among variance, bias, noise, and covariance, ensemble learners exhibit greater efficiency and stability than individual models. Practically, ensemble learning involves two primary stages: training the base classifiers and strategically integrating their outputs to achieve improved classification accuracy.

In traditional learning, a single classifier is responsible for all predictions, whereas ensemble learning relies on the collaboration of multiple classifiers. Ensemble methods can be categorized as homogeneous or heterogeneous. Homogeneous ensembles, such as RFs, consist of multiple instances of the same base learner (e.g., decision trees), whereas heterogeneous ensembles combine diverse classifiers such as Support Vector Machines (SVMs) and decision trees. The performance of an ensemble depends on both the diversity of its base classifiers and the method of combining their decisions. By employing multiple learning algorithms, ensemble learning consistently enhances the accuracy and robustness of classification frameworks, outperforming individual models. Common ensemble techniques include bagging, boosting, stacking, and voting, each contributing distinct mechanisms for improving model generalization and predictive power.

B. Bagging

Bootstrap aggregating, or bagging, is a significant ensemble learning technique recognized for its simplicity and effectiveness in reducing prediction variance without introducing bias. It operates by generating multiple bootstrap samples from the training dataset through random sampling with replacement and training separate base learners on each subset. The final prediction is obtained by aggregating individual outputs using majority voting [9]. Bagging is a parallel ensemble method since base learners are trained independently, ensuring computational efficiency and robustness. Key implementation challenges include determining the optimal number of base learners, the size of bootstrap samples, and the most effective fusion strategy for combining classifier outputs. The bagging process can be mathematically expressed as:

$$f(x) = \frac{1}{B} \sum_{B=1}^{B} f_b(x) \quad (1)$$

where $f_b(x)$ denotes each weak learner, and $\frac{1}{B}$ represents the total number of bootstrap samples.

C. Boosting

Boosting is an ensemble technique that combines multiple weak learners to form a strong predictive model. It iteratively trains weak classifiers on reweighted versions of the training data, giving higher importance to misclassified instances. The outputs of these classifiers are then aggregated through WMV to enhance overall accuracy. The AdaBoost algorithm exemplifies this approach by adaptively adjusting instance and classifier weights based on performance, thereby improving generalization. Despite its effectiveness, boosting presents challenges such as increased computational cost, susceptibility to overfitting with many iterations, and slower training compared to bagging due to sequential dependencies among models. Conceptually, boosting employs a sequential ensemble framework where each learner corrects the errors of its predecessors. The process can be expressed as [10]:

$$f(x) = \sum_t \propto_t h_t(x) \quad (2)$$

where $h_t(x)$ denotes each weak learner and $\propto_t$ represents its corresponding weight.

D. Stacking

Stacking is an ensemble technique that combines predictions from multiple base learners trained on the same dataset using a meta-learner, or a second-level model, to improve classification accuracy. The base classifiers are first trained to generate outputs that serve as input features for the meta-classifier, which then produces the final prediction. This approach effectively integrates the strengths of heterogeneous models to achieve superior performance. However, stacking introduces challenges, such as selecting the appropriate number and type of base models, managing computational complexity with large datasets, and mitigating risks of overfitting and high dimensionality, especially in multi-label classification. As a parallel ensemble method, stacking trains base learners independently, while fusion depends on the chosen meta-learning strategy. The stacking function is expressed as [11]:

$$f_s(x) = \sum_{i=1}^{n} a_i f_i(x) \quad (3)$$

A formal concept of stacking involves generating predictions from multiple models (m_1, m_2; m_3; …; m_n) to create an aggregated model, which is then utilized to make predictions on the test dataset. The objective of stacking is to enhance the predictive capability of a model. The main principle behind stacking is to "stack" the predictions of (m_1; m_2; m_3; …; m_n) through a linear combination of weights a_i; (I = 1; 2; …; n).

E. Voting

Voting is one of the most widely used ensemble strategies, combining predictions from multiple classifiers to determine the final output through a collective decision process. The method can be implemented as either unweighted (simple or majority voting) or weighted voting. In simple majority voting, the class receiving the highest number of votes from individual classifiers is selected. This approach requires no parameter tuning after model training and relies purely on consensus among classifiers.

Weighted voting, in contrast, assigns varying importance to classifiers based on their performance, giving higher weights to models or categories that demonstrate superior accuracy. The final decision is derived by aggregating these weighted outputs, often through averaging or weighted averaging, to improve robustness and precision. It has been shown that weighted voting-based ensembles outperform unweighted schemes by offering greater flexibility and more accurate class predictions [12].

F. Weighted Voting-Based Ensemble Methods

WMV has been successfully applied across diverse domains [13], including healthcare, environmental studies, intrusion detection, facial expression recognition, text mining, and software engineering. In medical diagnosis, ensemble classifiers combining bagging and multi-objective weighted voting have been used for heart disease detection with multiple base learners such as NB, Linear Regression, QDA, and SVM. Similarly, in credit scoring, a weighted voting approach integrated with supervised clustering has improved local accuracy and diversity among models. Other applications include evolutionary weighted voting for remote sensing and hybrid ensemble models for intrusion detection, combining

KNN and SVM with optimization techniques such as Particle Swarm Optimization (PSO) and WMV.

Beyond standard classification, WMV has demonstrated effectiveness in handling multi-label learning problems, with numerous studies emphasizing its performance advantages. Various optimization techniques, such as fuzzy sets, PSO, differential evolution, probabilistic frameworks, and Genetic Algorithms (GA), have been proposed for determining optimal classifier weights. The WMVE framework assigns weights based on validation accuracy, aggregating the most reliable classifiers to produce the final prediction [6]. Weighted ensemble models often employ base learners such as SVMs, neural networks, and decision trees. The present study adopts a WMV approach for image segmentation by combining classifiers, including KNN, RF, MLPC, Bayesian Network (BN), and XGB. Using soft voting with assigned weights, weak classifiers are aggregated into a robust ensemble that achieves superior segmentation accuracy and generalization compared to individual models.

G. Image Segmentation Using Ensemble Techniques

There has been growing interest in ensemble learning algorithms within ML due to their ability to improve prediction accuracy and reduce overfitting in complex models. The effectiveness of combining multiple models to enhance predictive performance has been demonstrated. Authors in [14] proposed a framework for improving accuracy by aggregating distinct models. Authors in [15] developed an algorithm for merging and evaluating segmentation outputs, while authors in [16] introduced a method to unify multiple segmentations for improved overall accuracy. Authors in [17] demonstrated that ensembles of networks with similar structures enhance the predictive capability of each model. Authors in [18] confirmed that ensemble-based segmentation outperforms traditional single-model segmentation methods. Authors in [19] proposed a fast ensemble learning technique that integrates deep convolutional networks with RFs, reducing training time with limited data. Authors in [20] advanced a model compression strategy that accelerates deep learning segmentation by combining diverse architectures to achieve real-time performance.

For aerial image segmentation, authors in [21] introduced an ensemble knowledge transfer technique involving the progressive fine-tuning and integration of multiple models to improve segmentation accuracy. Authors in [22] applied Fully Convolutional Networks (FCNs) within an ensemble framework for semantic segmentation of aerial images, demonstrating that integrating several networks yields superior performance. The success of model averaging in ML competitions, such as the Netflix Grand Prize [23], further attests to the efficacy of ensemble approaches. Authors in [24] developed an ensemble technique combining three deep learning–based portrait segmentation models, achieving a substantial improvement in segmentation accuracy. Among ensemble strategies, voting techniques are widely employed in both human and AI-assisted contexts to refine annotation precision, whether automatically or manually generated. Model voting represents a significant ensemble mechanism. In image segmentation, the predominant form, arithmetic voting, can be implemented as either hard or soft voting. In this method, each pixel is classified based on a majority rule applied to its own predicted values, without considering neighboring pixel relationships.

To construct and combine a large set of classifiers, either weak or strong, ensemble methods are often preferred over reliance on a single learning algorithm. The predictive contribution of each classifier can differ, making it advantageous to weight votes according to classifier performance rather than treating them equally. The effectiveness of any ensemble technique depends heavily on the appropriate selection of base learners and the combination strategy used to derive the final decision. The WMVE approach implements this by assigning each classifier a weight proportional to its validation performance, producing a final decision based on the most heavily weighted votes. Building on this foundation, the present research evaluates ML algorithms and ensemble strategies to determine the most effective methodology for image segmentation. The proposed framework, grounded in the WMV paradigm, comprises four main phases: (1) Preprocessing, involving filtering operations to ensure accurate segmentation and reliable feature extraction, (2) Training, where classifiers are trained on the training subset of the data, (3) Weight determination, in which classifier weights are computed using the validation set, with weight adjustments based on each classifier's ability to predict class labels correctly, and (4) Integration, which combines classifier outputs according to their respective weights, yielding the final segmentation decision through WMV across all predicted classes.

III. RESEARCH METHODOLOGY

A. Base Classification Model

To achieve optimal performance, the careful selection of base learners and the accurate determination of the ideal number of weak classifiers are essential factors. It is not guaranteed that increasing the quantity of classifiers will consistently result in better outcomes, since the evaluation of the classifier's performance is based on the average of all the combined classifiers. Additionally, a larger number of integrated classifiers will prolong the training process. However, the majority voting classifier, consisting of five tree-based algorithms, strikes a harmonious balance between computational efficiency and effectiveness.

B. K-Nearest Neighbors (KNN)

When there is limited or no understanding of the data distribution, the KNN algorithm is widely regarded as the cornerstone of classification algorithms and is typically used as the first choice. The KNN algorithm falls under the category of supervised learning. In KNN, a sample is classified by analyzing the k nearest training instances and taking into account the majority of these neighbors. This is followed by assigning a category to a new sample. The Euclidean distance is frequently employed as a metric in the KNN algorithm [25]. It is important to acknowledge the significance of two key aspects in KNN: the choice of k and the distance function.

C. Multilayer Perceptron (MLP)

The MLP is classified as an artificial neural network and is commonly used for prediction tasks. The role of the MLP is important in helping to define and determine significance. The MLP is composed of multiple layers:

$$a^{m+1} = f^{m+1}(w^{m+1}\, a^{m+1} + b^{m+1}) \quad (4)$$

where w is the weight vector, b is the bias vector, a is the output vector, and m is the layer rank. Within each layer, the inputs are aggregated, followed by the calculation of the weight of each node. These values are then used as input for the subsequent layer through a node.

D. Bayesian Belief Networks

Bayesian belief networks are probabilistic graphical models that aim to overcome the limitations of NB by addressing the assumption of attribute independence. By representing conditional independence among variables, this approach aims to overcome the limitations of NB, specifically the assumption that X is independent of Y given Z. This is expressed as: $P(X \mid Y, Z) = P(X \mid Z)$. The Bayes network consists of variables with associated probabilities, which may represent root node probabilities or joint probabilities obtained by connecting nodes:

$$Join\ prob. =$$

$$p(X_1, \ldots., X_n) \prod^n p(x_i(Parents(x_i)) \quad (5)$$

E. Random Forest (RF)

Decision trees [26], a widely used and powerful tool in ML, often suffer from low accuracy due to their high variance and low bias. To address this issue, RFs were introduced as a method to improve model performance by merging a set of decision trees trained on diverse parts of the same dataset. This method was originally proposed as a combination of bagging with the random selection of variables node by node. In the bagging process, a random subset of the training set is repeatedly chosen (with replacement) and used to train individual trees. This integration of techniques reduces variance and improves the model's performance as a whole. For each value of m from 1 to M:

- Let us consider a sample subset, where n training data points from X and Y are randomly replaced. Let us call this subset X_m and Y_m.
- Let us train a classification tree f_m on X_m, Y_m using a modified tree learning algorithm.
- At each stage of the feasible candidate division, using a modified tree learning algorithm, the system randomly selects a sample from the attribute set. This procedure, known as feature bagging, is illustrated in Algorithm 1.

Algorithm 1: Find spl^* (the best split) that divides f_t among a random subset of D ≤ p input variables.

```
1. Function FindBestSplitRandom (f_t D)
2. Δ = −∞
3. Draw D random indices J_d from 1,….P
4.     For d = 1, …, D do
5.           Find the best binary split
spl* J_d defined in X_jd
6.                 if Δ_i (spl* J_d) > Δ then
7.                       Δ = Δ_i (spl* J_d, t)
8.                       spl* = spl* J_d
9.                             end
10.   end
11. return spl*
12. End
```

F. Extreme Gradient Boosting (XGB)

XGB is a scalable and efficient tree-based gradient boosting algorithm designed for parallel computation. It begins by training an initial decision tree to make predictions and iteratively improves performance by increasing the influence of misclassified samples during subsequent training rounds. This process continues until the optimal number of iterations is reached, resulting in a highly accurate and generalized model. XGB is versatile, capable of handling various data types, complex relationships, and diverse distributions between predictors and responses. Its strength lies in its flexibility, extensive hyperparameter control, and proven effectiveness across numerous ML applications. The model output is expressed as a weighted sum of decision trees, where each tree contributes according to its learned weight and parameterized structure:

$$y_{xgboost} = \sum_j\ \alpha_j c_j(\underline{x_i})) = \sum_j\ \alpha_j f(\underline{x}\ ; \underline{\theta_i}) \quad (6)$$

where $\alpha(j)$ is the weight of the classifier, and C_j is the parameterized function denoted by $f(\cdot)$ for the decision tree. The splitting tuples for the i^{th} tree are grouped in the vector θ_i.

G. Majority Voting Ensemble Model

The ensemble method has emerged as a popular trend in ML. This method involves analyzing a set of classifiers and carefully selecting a subset to create an ensemble model. By combining the predictions of these individual classifiers, superior performance is achieved compared to using a single classifier. In order to build an effective ensemble, the individual learners should possess a combination of accuracy and diversity. There are multiple algorithms available for ensemble learning, but the most widely utilized approaches for numeric and nominal outputs are averaging and voting [27]. Different types of voting methods are majority voting, plurality voting, weighted voting, and soft voting. For majority voting, class probability outputs are generated by individual classifiers. There are two categories under majority voting: simple majority voting and WMV.

H. Simple Majority Voting Ensemble

Majority voting is a commonly employed and direct approach to combining predictions from various classifiers. There are multiple methods for implementing majority voting, but the prevalent approach typically entails assigning an instance to the class that receives the highest number of votes from the base classifiers. In this particular voting scheme, all classifiers hold an equal weight in terms of their votes, which is set to 1. Let $E = \{C_1, C_2, ..., C_N\}$ represent an ensemble, and C_t denotes a classifier where t varies from 1 to n. A decision made

by the t^{th} classifier is denoted by $d_{t,j} \in \{0,1\}$. In this case, variable j ranges from 1 to k, where k represents the total number of classes. If the t^{th} classifier chooses class C_j, then the decision $d_{t,j}$ is equal to 1; otherwise, it is 0. The output of the ensemble can be summarized as:

$$max_{1 \le j \le k} \sum_{t=1}^{n} d_{t,j} \quad (7)$$

For ensembles, the individual classifiers often have varying levels of performance. Therefore, it is not ideal to treat them all equally when combining their results. Instead, one of the most important parts of an ensemble method is assigning weights to each classifier, which determines the weight of each classifier's performance. The performance of the ensemble depends on determining these weights.

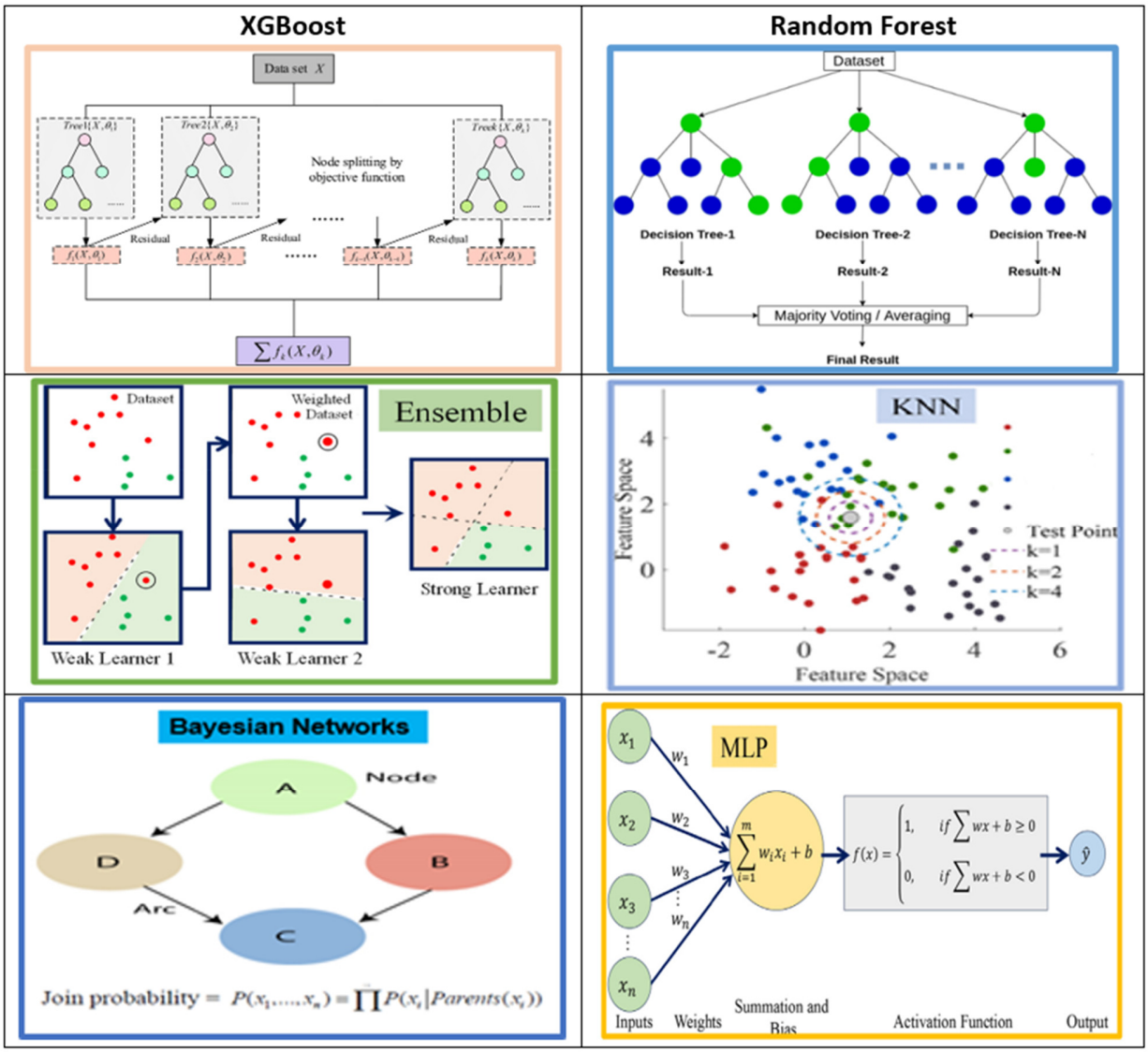

Fig. 1. Schematic representation of the ML classifiers for the proposed method.

I. WMVE Framework

Authors in [28] introduced the WMV algorithm in 1994. The outcome of this algorithm is determined by a single learner, where the decision is made by taking weighted majority votes provided by the individual algorithms. This algorithm was demonstrated to be resilient against data errors. In particular, when algorithm A commits a maximum of m errors, the WMV algorithm does not exceed c $(\log|A| + m)$ errors on the identical sequence, where c represents a constant value. Each algorithm in the pool is assigned a non-negative weight, initially set to one unless otherwise specified. Predictions are generated by the algorithm by comparing the cumulative weights linked to each class, predicting the class with the highest total weight (in case of a tie, an arbitrary prediction is made). In instances where the weighted majority algorithm results in error, the algorithms that align with the error have their weights multiplied by a constant value, β, where $0 < \beta < 1$. Based on the weights assigned to each classifier, these weights serve as the main factors in

determining the predicted class labels of test set instances. The class prediction for a given instance is determined by aggregating the weighted votes across all classes and selecting the class with the maximum total weighted votes, as defined by:

$$max_{1\le j\le k} \sum_{t=1}^{n} w_t d_{t,j} \quad (8)$$

The procedure of WMV is illustrated in Algorithm 2.

Algorithm 2: The weighted majority algorithm
Input $0 < \beta < 1$is the learning rate of the algorithm.
Initialize weights $W_i = 1$ for all classifiers i
For $i = 1$ to Ndo
3. Receive a set of predictions $\{x_1, x_2, \dots, x_n\}$
4. Compute $m_0 \leftarrow \sum_{i=1}^{N} x_{t,i} = 0\, W_i,\ m_1 \leftarrow \sum_{i=1}^{N} x_{t,i} = 1\, W_i$
5. Predict $\hat{y} = 0$ if $m_0 > m_1,\ \hat{y} = 1$ otherwise
6. Receive true outcome y
7. If $(\hat{y} \neq y)$then $\forall x_i \neq y, W_i \leftarrow \beta \cdot W_i$
End for

J. Proposed Method

The proposed WMVE method introduces a simple and fast weighting scheme based directly on CV performance, in contrast to the heuristic or iterative optimized schemes of most conventional weighted voting ensembles. There is no meta-optimization or extra learning stage involved for computational overhead, while strong predictive performance is achieved. Another novelty introduced by this framework lies in its formulation, particularly suitable for image segmentation tasks through region-based evaluation metrics incorporated at the model selection phase, rather than using only a pixel-level accuracy measure. The proposed method combines heterogeneous ML classifiers having complementary decision characteristics within a single adaptively tunable voting strategy, thus making the resultant system effective as well as generalizable over different segmentation scenarios. Ensemble learning is the combination of multiple classifiers or base learners to attain a dependable predictive model that cannot be attained by a single algorithm, as shown in Figure 2. In WMV, more weight is given to those classifiers that are found reliable on validation data rather than using simple conventional majority voting. The selection of base learners and aggregation method plays a key role in such ensemble strategies. A different weight proportional to its validation accuracy is assigned to each classifier within the proposed WMVE framework, and the final segmentation decision is made based on summing up all weighted votes from every classifier, thereby reducing individual model error while providing a practical yet effective solution for achieving high-performance image segmentation. Figure 1 illustrates a detailed block diagram depicting the use of four base learners, namely KNN, BN, RF, XGB, and MLP. The weights of each classifier were chosen arbitrarily to obtain a total of 100, and then the probability of classification was used to determine the final ensemble result.

K. De-Noising Filtering

To ensure accurate image segmentation and feature extraction, the initial step involves filtering. This crucial phase lays the foundation for obtaining objective and dependable data for further analysis. In this work, the input image underwent preprocessing using the Gaussian smoothing filter and the median filter. The Gaussian filtering involves calculating the weighted average pixel values within a specific neighborhood. The Gaussian filter function is defined as:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (9)$$

where x and y are the horizontal and vertical coordinates of the image pixel, respectively, and σ is the standard deviation.

To perform median filtering on a digital image, the gray value of each pixel is obtained by applying the median gray value to all pixels inside the surrounding window. The median filtering can be defined as:

$$g(x,y) = median_A\{f(x,y)\} \quad (10)$$

where $g(x,y)$ is the output of median filtering, $\{f(x,y)\}$ represents the original image, and A is the size of the filtered neighborhood.

The Gaussian is a linear filter, often used to blur the image or to reduce noise. In the present study, Gaussian and median filters were employed for edge detection. The Gaussian filter reduces blurred edges and contrast, while the median filter reduces noise in an image.

L. WMVE Classifier

The effectiveness of an ensemble technique is maximized when both the elementary learners and the method of combining for the final decision are appropriately selected. In the WMVE method, each trained classifier is assigned a distinct weight according to its performance in classifying the validation set. This weighting is determined by the classifier that performs most effectively. The final prediction for each occurrence is determined through the use of weighted votes. This approach comprises three distinct phases. The first phase involves the training of classifiers utilizing training data. From the complete dataset, three subsets are generated: training, validation, and test sets. In the second step, weights of the classifiers are derived from the validation set analysis. In this step, each classifier provides its own decision, with the weights updated based on the evaluation of decisions by class label for a particular instance. The weights for each classifier are determined as:

$$w_c = \frac{a_{c_i}}{\sum_{n=1}^{n} a_{c_i}} \quad (11)$$

where w_c is the weight for classification, while a_c represents the classification accuracy. The total classification accuracy is given by:

$$an = \sum_{n=1}^{n} a_{c_i} \quad (12)$$

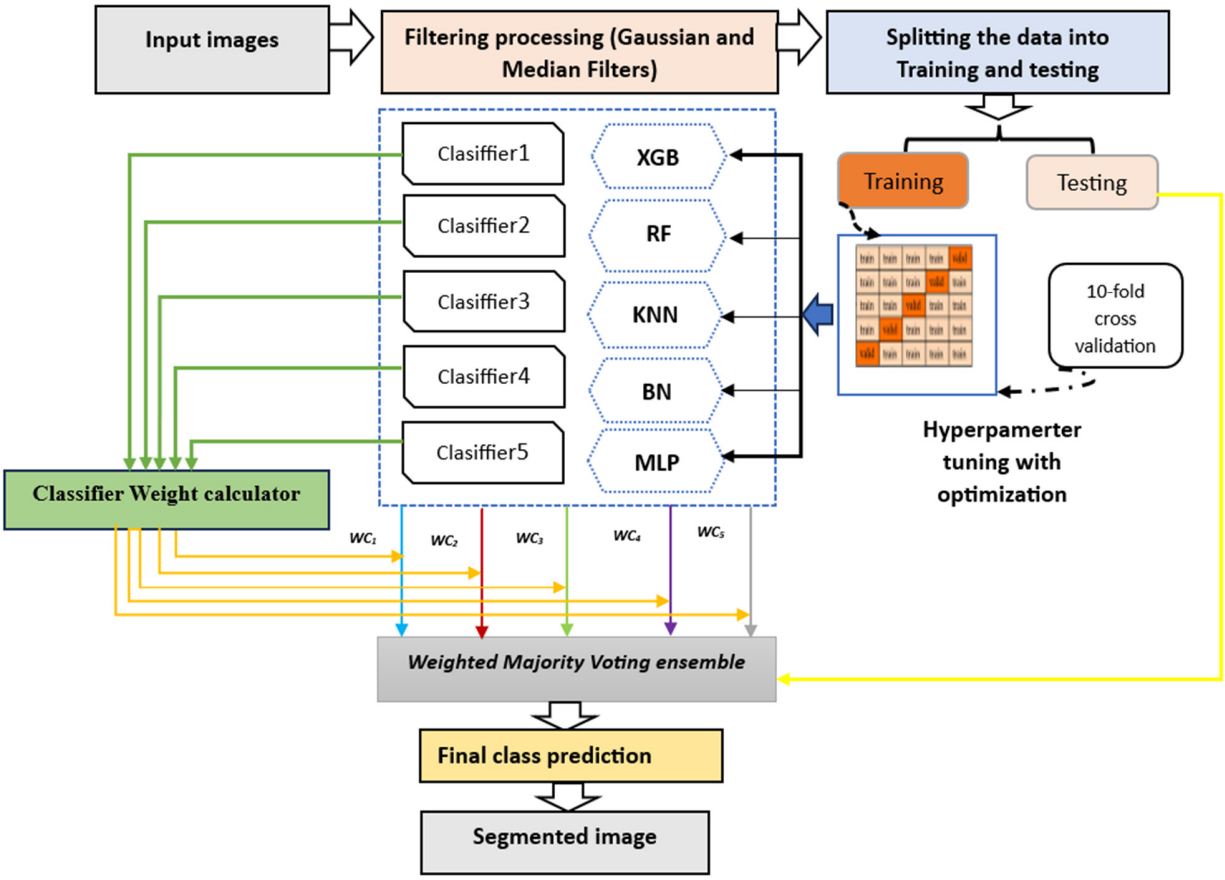

Fig. 2. Workflow of the proposed ensemble vote-based model.

The weighting strategy is based on validation accuracy. The latter remains stable and represents the reliability of a classifier. Validation accuracy helps keep the model computationally efficient by avoiding any extra optimization stage or meta-learning component to the ensemble. F1 score, entropy, and inferred weights through meta classifiers can be an alternative measure of precision; however, these approaches add to computational complexity, requiring further training and tuning, thus constraining scalability and reproducibility. In contrast, validation accuracy provides a strong interpretable criterion that correlates well with region-based performance measures in the case of image segmentation. Analysis, therefore, captures a simple yet deliberate tradeoff between simplicity and effectiveness that it can achieve competitive segmentation performance without using meta learners or iterative weight optimization schemes. The third step involves the integration of outcomes from multiple classifiers, considering their respective weights. The ensemble's definitive result is determined by selecting the prediction that received the highest number of votes, along with the corresponding weight assigned to each classifier. The WMV technique calculates classifier weights using training datasets with 10-fold CV.

M. Hyper-Parameter Optimization Through Random Search

Every ML model comprises a set of hyperparameters that require optimization for each feature subset. To fine-tune the hyperparameters associated with the aforementioned base classifiers, the study employs the random search approach [29]. This method involves testing different random combinations of hyperparameters in each iteration to evaluate the effectiveness of the model. Through a series of iterations, the function identifies the combination that yields the best result, surpassing the limitations of the grid search technique. One advantage of using random search is its ability to produce results more quickly than grid search. Additionally, it ensures that the proposed model remains uninfluenced by predetermined value sets chosen by the user. The ML estimators implemented in this research are RF, XGB, KNN, MLP, and BN. The decision to limit the selection to five models stems from the intention to maintain an odd number of models, which is advantageous for the WMV process at the second level. An increase in the number of odd models enhances the probability of obtaining a greater number of correct votes, although it also increases complexity. Consequently, this work proposes the selection of five models as a balanced quantity. To effectively evaluate performance, a 10-fold CV is employed. CV serves as a statistical technique for assessing the performance of ML models and is considered an effective approach for smaller datasets. The general procedure for CV is outlined as: (i) randomly shuffle the dataset, (ii) partition it into k-groups, (iii) for each distinct group, utilize that group as the testing subset while employing the remaining groups as the training subset, fit a model on the training subset, evaluate its performance on the

testing subset, record the evaluation score, subsequently discard the model, and finally, (iv) aggregate the model's metrics by summarizing the collected evaluation scores. This work employed preprocessing techniques to identify the optimal features for each ML algorithm utilized (i.e., RF, XGB, KNN, MLP, and BN) in the initial stage of the proposed approach. In the subsequent stage, the study applied a WMV strategy to rank the features. A random search was conducted for the ML algorithms to determine the most effective hyperparameters. The various hyperparameters for each model used are presented in Table I. The designation N/A signifies that the model utilizes the default hyperparameter settings without any additional optimization.

TABLE I. FINAL HYPERPARAMETER CONFIGURATIONS SELECTED VIA GRIDSEARCHCV

Estimator/Model	Hyperparameter	Value
XGB	Number of estimators (n_estimators)	200
	Learning rate (learning_rate)	0.1
MLP	Activation function (activation)	ReLU
	Learning rate method (learning_rate)	Adaptive
	Solver (solver)	Adam
RF	Number of trees (n_estimators)	100
	Maximum depth (max_depth)	50
	Minimum sample split (min_samples_split)	2
	Minimum samples per leaf (min_samples_leaf)	1
	Split criterion (criterion)	gini
	Maximum features (max_features)	64
KNN	Number of neighbors (n_neighbors)	5
	Distance metric	Euclidean
BN	NA	NA

The model evaluation is based on several metrics, including fitness values, Peak Signal-to-Noise Ratio (PSNR), and Feature Similarity Index (FSIM), and the set of evaluation metrics commonly used in segmentation analysis, such as SC, VoI, PRI, GCE, and BDE. These metrics serve as evaluation indices for the test image. Higher scores in PSNR, FSIM, SC, and PRI indicate superior results, whereas lower values in VoI, GCE, and BDE signify better segmentation [30]. The PSNR measures the ratio between signal and noise for the initial image and the image resulting from segmentation. The PSNR index is determined as:

$$PSNR = 20\ log_{10}\left[\frac{255}{RMSE}\right](dB) \quad (13)$$

$$RMSE = \sqrt{\frac{1}{M*N}\sum_{i=0,j=0}^{M-1,N-1}\ [I(i,j) - K(i,j)]^2} \quad (14)$$

where *M* and *N* are the size of the image, *I* is the input image, and *K* is the segmented image.

Similarly, FSIM is employed to assess the structural similarity between the input and the segmented image. FSIM is defined as:

$$FSIM = \frac{\sum_{X\in\Omega}\ S_L(X)P_{Cm}(X)}{\sum_{X\in\Omega}\ PC_m(X)} \quad (15)$$

where Ω represents the entirety of the image, while *SL*(*x*) denotes the resemblance of the segmented images produced through the multilevel thresholding. When evaluating color RGB images, the FSIM metric can be defined as:

$$FSIM = \frac{1}{\theta}\sum_{\theta}\ FSIM\ (x^{\theta}, y^{\theta}) \quad (16)$$

where x^{θ} and y^{θ} are the initial image and the image resulting from segmentation, respectively, and θ is the channel number. The SC quantifies the degree of overlap between the regions identified in the output from the segmentation process and the regions identified in the ground truth. The higher SC value indicates better and more reliable segmentation. SC can be calculated as:

$$S(S' \to S) = \sum_{R\in S}\ |R|.max_{R'\in S'}O(R,R') \quad (17)$$

where *O* represents the overlap between regions *R* and *R*'. This overlap can be determined by the resolution in the image (*N*) and the pixel count in region *R* (|*R*|), as:

$$O(R,R') = \frac{|R\cap R'|}{|R\cup R'|} \quad (18)$$

The PRI measures the similarity between the outcomes produced by the algorithm and the expected results. Multiple manually segmented images are used in this approach by taking into account various outputs. This approach is more reliable as human perception is also taken into consideration during verification. The weighting of pixel pairs in the PRI is flexible and non-uniform. The choice of the labeled image set is objective and is based on the measured variability in the manual ground truth. The PRI value is given by:

$$PR\ (S,G_k) = \frac{2}{N(N-1)}\sum_{i,j\ i<j}\ (p_{ij}^{C_{ij}}(1-p_{ij})^{1-C_{ij}}) \quad (19)$$

where *S* is the segmentation generated by the algorithm, x_i is the pixel in k^{th} manually segmented image with labels l_i^S and $l_i^{G_k}$, *N* is the PRI ranging from 1 to 0, and c_{ij} is the Boolean function which determines if l_i^S is equivalent to $l_i^{G_k}$. Additionally, the value of p_{ij} corresponds to the anticipated outcome obtained from a Bernoulli distribution for the pair of pixels. Higher values on the PRI metric indicate a significant resemblance between segmented images and their realistic depiction.

The VoI metric is primarily introduced to facilitate the comparison of clusters. It quantifies the extent of missed and additional information between two sets of images, utilizing their mutual information and marginal entropy:

$$\text{VoI}\ (S,G_k) = H(S) + H(G_k) - 2I(S,G_k) \quad (20)$$

where the entropy, $H = -\sum_{i=1}^{n}\frac{n_i}{n}\ log\frac{n_i}{n}$, is calculated by considering the number of points in a cluster, represented by n_i. The mutual information between two clusters, known as the pronoun *I*, is defined as:

$$I(S,G_k) = \sum_{i=1}^{c}\ \sum_{j=1}^{c}\ \frac{n_{i,j}}{n}\ log\frac{n_{i,j}}{n}\frac{n_i}{n}\frac{n_j}{n} \quad (21)$$

The VoI quantifies the number of points that intersect between cluster *i* of *S* and cluster *j* of G_k, denoted as $n_{i,j}$. This measure functions as a distance, meaning that a smaller value

indicates a closer alignment between the obtained segmentation and the authentic representation.

The GCE operates under the assumption that one of the segmentations is a more detailed version of the other and requires that all local enhancements align in the same direction. To quantify the level of error at each pixel x_i, a measurement is established as:

$$E(S, G_k, x_i) = \frac{|R(S,x_i) \setminus R(G_k,x_i)|}{|R(S,x_i)|} \quad (22)$$

The GCE value depends on local refinements and their orientations, with the set difference operator (\) used to determine the cardinality of the set. Furthermore, the set R (S, x_i) denotes the set of pixels that belong to the region of segmentation S, including x_i:

$$\mathrm{GCE}(S, G_k) = \frac{1}{n} min \left(\sum_{i=1}^{n} E(S, G_k, x_i), \sum_{i=1}^{n} E(G_k, S, x_i)\right) \quad (23)$$

As the value of GCE approaches zero, the level of accuracy in the segmentation S is higher than that of ground truth G_k.

The BDE metric is a unique method used for evaluating quantitative segmentation. It specifically focuses on distance-based features, making it highly sensitive to the sensitivity of the model, and is reflected in its accuracy in matching the object contours detected by automated segmentation with those same object contours in the ground truth (G_T) image. BDE index is calculated by considering an arbitrary point x_i from the segmented image (S) and finding the minimal Euclidean distance between x_i and all points of the corresponding object in the G_T image (G_k). The distribution signature D is obtained by summing these distances over all points in S. Finally, the BDE index is defined as:

$$\mathrm{BDE}(S, G_k) = \frac{1}{2}(D_S^{G_k} + D_{G_k}^{S}) \quad (24)$$

Once again, a smaller BDE value suggests that the segmentation of S is of high quality in relation to the ground-truth segmentation G_k.

IV. RESULTS AND DISCUSSION

Filtering is the preliminary step in image segmentation and feature extraction. It is critical for acquiring objective and reliable data for subsequent analysis. The present study employs Gaussian smoothing and the median filter techniques. The proposed framework also incorporates random search for hyperparameter tuning. This comprehensive approach leads to optimal parameter configurations within the specified range. Although this study focuses on a specific image dataset, the proposed frame is also applicable to other datasets. In addition, this study is based on the assumption of computational feasibility in the Google Colab environment. Recognizing this practical limitation was crucial for the successful execution of experiments; however, it may impose restrictions on the scalability to larger datasets or more intricate models. The COCO 2014 dataset was used to demonstrate the effectiveness of the proposed framework and assess the performance of the image segmentation technique. The dataset contains 80 object categories and approximately 40,000 training images [31]. In addition, the Berkeley Segmentation Dataset (BSDS500) [32] was employed for image segmentation evaluation, and the PASCAL VOC 2012 [33] dataset was used for object detection and segmentation tasks.

A. Segmentation Performance Analysis

The proposed method utilizes WMV, which effectively integrates the outputs of individual classifiers by taking their respective weights into account. This approach results in a more robust and optimal classifier, leading to enhanced segmentation outcomes. The qualitative results indicate that the proposed method provides accurate segmentation compared to individual models, other established ensemble techniques such as bagging, boosting (including AdaBoost, XGB, and gradient boosting), and stacking. Furthermore, the findings reveal that the proposed method successfully segments images into a reasonable number of clusters, accurately delineates coherent regions, detects objects, and addresses issues related to under-segmentation. Figure 3 presents five examples of segmentation results generated by five base classifiers: RF, KNN, BN, XGB, MLP, and WMVE. The segmentation contours produced by the proposed method closely resemble the manual annotations and outperform the results obtained by the other five methods. Test images were employed to evaluate the effectiveness of the proposed method. The PSNR values of the segmented results are depicted in Figure 4. It is observed that higher PSNR values indicate more accurate and improved segmentation. The results demonstrate that the proposed method outperforms individual models. To better understand the differences between algorithms, a statistical analysis was conducted on the experimental results using FSIM. The statistical analysis results, as portrayed in Figures 5 and 6, indicate that the proposed framework achieved better performance compared to individual algorithms. In addition, the performance of the proposed method was compared with other popular ensemble methods such as RF (bagging), AdaBoost, Gradient Boosting, and stacking. Figure 7 illustrates the comparison of execution time for these methods. The results indicate that the specific methods were significantly faster than the proposed ensemble method. However, when compared with the stacking approach, the execution time of the proposed approach was significantly lower.

The proposed WMVE framework introduces extra computational overhead relative to the individual base models. However, this overhead is minimal and remains within acceptable limits. As displayed in Figure 7, WMVE runs much faster than stacking-based ensembles because there is no training or inference of any meta-learner; only a weighted aggregation of model outputs is performed. The computational complexity in WMVE grows linearly with respect to the number of base models present, thus giving an excellent tradeoff between gain in performance and added computational cost. Resolution-dependent operations are not introduced by WMVE beyond those already required by the base models; hence, scalability towards high-resolution images seems mainly determined by underlying architectures.

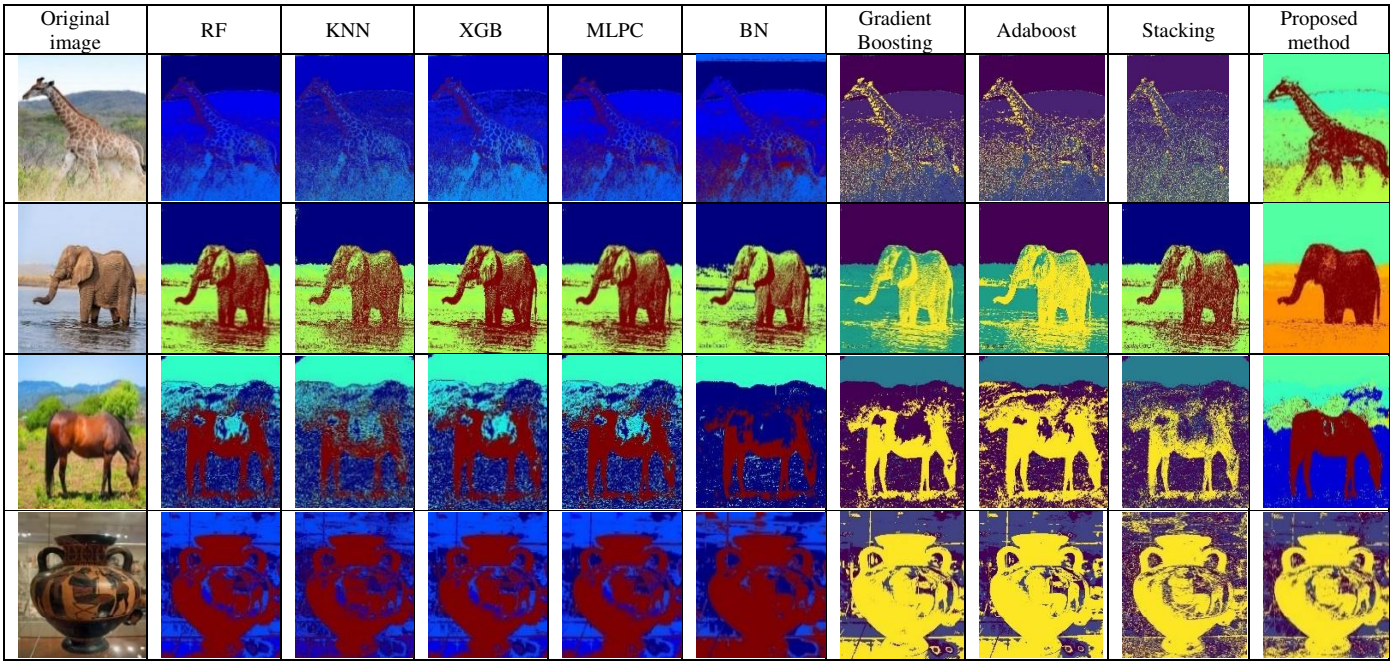

Fig. 3. Qualitative image segmentation results obtained using different methods.

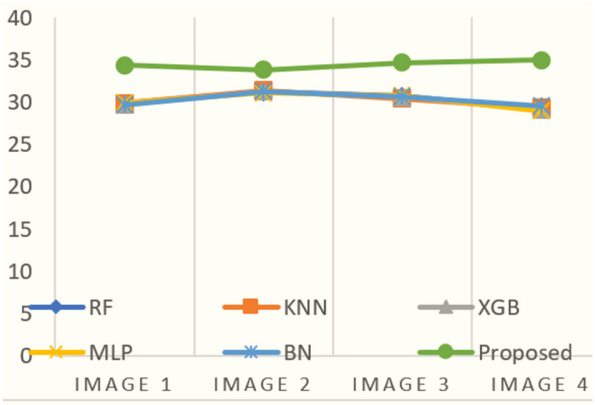

Fig. 4. Comparison of PSNR values for different algorithms.

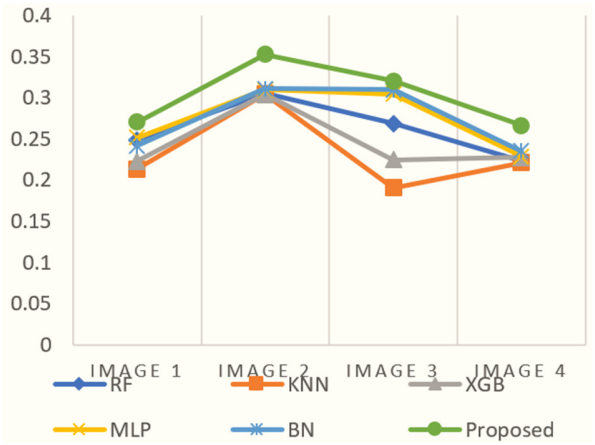

Fig. 5. Comparison of FSIM values for different algorithms.

The analysis in Table I indicates that performance outcomes vary depending on the image being analyzed. However, the proposed method surpasses the performance of other algorithms, consistently yielding superior results. The proposed approach demonstrates excellent segmentation ability, highlighting the distinct regions within the image. In addition, the proposed method provides favorable results for SC, PRI, VOI, BDE, and GCE, demonstrating satisfactory performance.

Table II compares the evaluation criteria of the proposed ensemble method with individual methods as well as other leading ensemble techniques, such as bagging, gradient boosting, XGB, and stacking, for each test image. The results indicate that model performance varies based on the image; however, the proposed method consistently outperforms the other algorithms. The proposed technique obtained well-defined segmentation results, with distinct regions of the image being clearly discernible, and achieved favorable values for SC, PRI, VoI, BDE, and GCE.

The results of the statistical analysis, presented in Figures 4-6 and Table II, indicate that the quality of the segmented images varies across the different techniques. The proposed WMV-based ensemble method demonstrates favorable performance in cluster quality measures based on the experimental results. The results across multiple reference and real images demonstrate the efficiency of the proposed method in terms of accuracy and reliability.

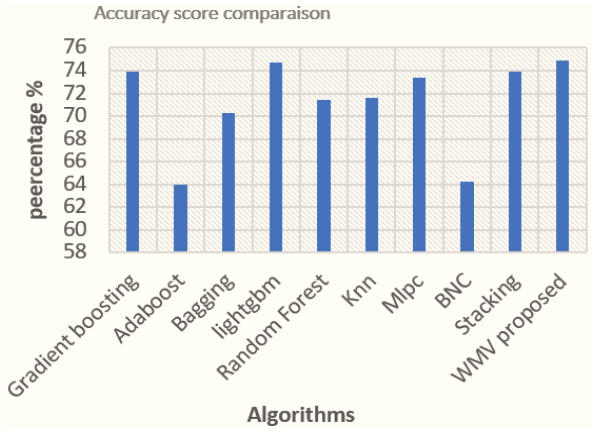

Fig. 6. Accuracy for different algorithms.

It is often beneficial to use multiple metrics for a comprehensive understanding of the model's performance. Choosing appropriate metrics leads to more generalized and reliable results from classification models and improves usability in real-world applications.

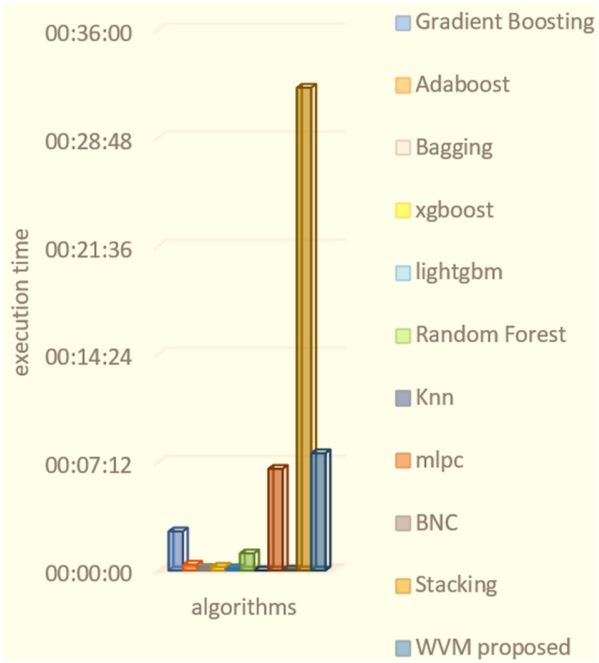

Fig. 7. Execution time for different algorithms.

TABLE II. STATISTICAL ANALYSIS OF IMAGE SEGMENTATION PERFORMANCE WITH OPTIMAL VALUES

Images	Metrics	RF	KNN	XGB	MLP	BN	Gradient boosting	Adaboost	Stacking	Proposed method
(a)	SC	0.4835	0.5319	0.5386	0.4822	0.5040	0.4816	0.4838	0.5456	0.7315
	PRI	0.4611	0.4761	0.4723	0.4633	0.4647	0.4605	0.4620	0.4733	0.4966
	VOI	2.14573	2.2665	2.1978	2.1835	2.3111	2.1475	2.1501	2.1497	1.4503
	GCE	2.0594	2.1618	2.1319	2.0841	2.1586	2.0590	2.0625	2.1072	1.4340
	BDE	165.4223	152.0386	163.9220	165.7995	154.5132	166.9375	166.0263	166.2690	102.7594
(b)	SC	0.5393	0.5651	0.5411	0.5264	0.5334	0.5504	0.5231	0.5581	0.8009
	PRI	0.4628	0.4660	0.4625	0.4612	0.4635	0.4648	0.4604	0.4653	0.4953
	VOI	1.7451	1.7163	1.7454	1.7613	1.7700	1.6940	1.7489	1.7424	0.8035
	GCE	1.7573	1.7502	1.7554	1.7735	1.7724	1.7213	1.7790	1.7692	0.8549
	BDE	179.2637	174.4048	177.9586	177.7745	185.9243	182.8902	175.3666	183.1345	158.2712
(c)	SC	0.5395	0.5796	0.5636	0.5143	0.5184	0.5155	0.4992	0.5689	0.5474
	PRI	0.45061	0.4613	0.4567	0.4447	0.4418	0.4464	0.4540	0.4602	0.4966
	VOI	1.5628	1.6916	1.5850	1.5460	1.5157	1.5223	1.5620	1.6780	0.9940
	GCE	1.56176	1.6417	1.5550	1.5609	1.5347	1.5455	1.5851	1.6624	0.9240
	BDE	98.4421	136.0948	159.5605	86.2623	96.9323	91.7504	91.4549	121.7714	99.9302
(d)	SC	0.3368	0.3507	0.3373	0.3313	0.3363	0.3236	0.3234	0.3676	0.4685
	PRI	0.4353	0.4389	0.4308	0.4312	0.4185	0.4268	0.4276	0.4405	0.4860
	VOI	1.3164	1.3244	1.3015	1.3465	1.3944	1.3176	1.3151	1.3529	0.6864
	GCE	1.4326	1.4652	1.4374	1.4784	1.5228	1.4430	1.4350	1.5062	0.6964
	BDE	159.0995	153.2184	180.6104	171.7398	119.1400	145.787	143.8531	132.7628	145.4586

Additional evaluation metrics, such as balanced accuracy, precision, sensitivity, specificity, F1 score, and Matthews Correlation Coefficient (MCC), were included to improve the interpretability of the ensemble results. These metrics provide insights into how the ensemble method derives its predictions and the contribution of individual features or models to the final decision. Figure 8 illustrates the comparison of different evaluation metrics for different algorithms. It can be observed that the proposed WMVE framework improves performance significantly.

B. Experimental Results

The experiments were conducted using the COCO, BSD500, and VOC datasets. These datasets consist of numerous ground truth segmentations for each image, which helps perform the segmentation of multiple performance indices. The overall performance index for the segmentation is calculated by averaging these performance indices. Figure 9 compares the segmentation performance of the proposed method with other methods, including classical techniques such as GMM [34] and recent deep learning-based methods such as Differentiable Feature Clustering (DFC) [35], WNet [36], DeepCluster [37], and DEM-NetS [38], on the COCO dataset.

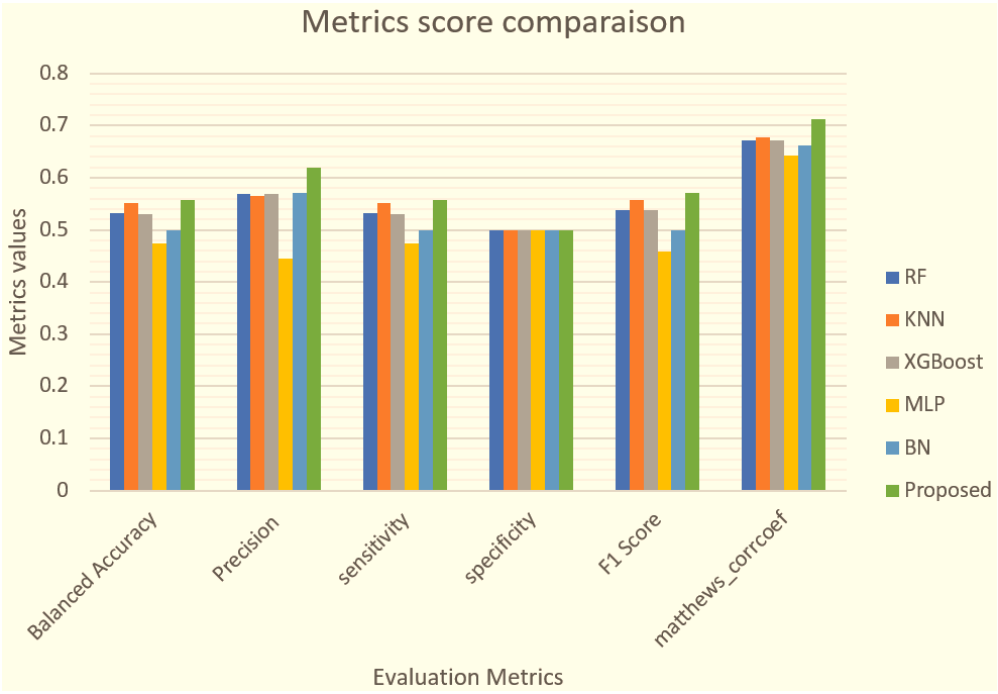

Fig. 8. Comparison of different evaluation metrics for different algorithms.

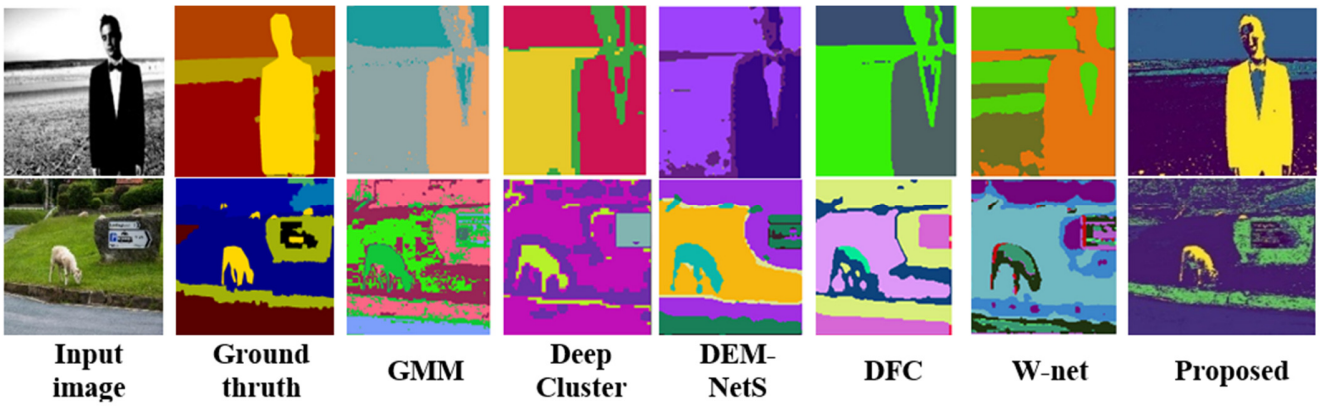

Fig. 9. Results obtained using different unsupervised image segmentation methods.

TABLE III. QUANTITATIVE PERFORMANCE RESULTS FOR BSDS500 DATASET

Method	SC↑	PRI ↑	BDE↓ (px)	GCE ↓
Backprop [39]	0.58	0.82	10.4	0.19
NCuts[40]	0.53	0.78	17.2	0.23
CAE-TVL[41]	0.61	0.85	8.3	0.16
Mean shift [42]	0.56	0.80	13.5	0.21
gPb-owt-ucm [33]	0.59	0.83	9.5	0.21
HED + UCM [43]	0.61	0.85	8.7	0.17
RCF + UCM [44]	0.62	0.86	8.1	0.15
DeepLab-CRF [45]	0.63	0.87	7.9	0.14
BDCN + hierarchical grouping [46]	0.64	0.88	7.3	0.13
U-Net (BSDS-trained) [47]	0.61	0.85	8.5	0.16
SegNet [48]	0.60	0.84	8.9	0.18
Proposed WMVE	0.64	0.71	14.4	0.16

Table III provides a quantitative comparison between the proposed WMVE and some representative classical, clustering-based, and deep learning segmentation approaches using SC, PRI, BDE, and GCE metrics. The proposed WMVE obtains an SC score of 0.64, which is excellent for deep learning methods such as BDCN with hierarchical grouping. Moreover, WMVE

achieves a very competitive GCE value of 0.16, similar to several CNN-based approaches. The proposed model achieved a significantly lower PRI with BDE. This trend indicates a clear trade-off between model complexity and accuracy. The method still achieves good region coverage and consistency scores. It also outperforms many non-deep ensemble strategies in terms of lower training complexity and computational requirements.

The results reported in Table III show that, even if using traditional ML classifiers, the proposed method achieves a level of performance comparable to the best deep learning-based methods. This is because of an effective WMVE strategy combining complementary information from several models, leading eventually to good segmentation. Unlike deep learning, which generally requires huge labeled datasets and runs on massive computational resources, the strong performance by/of this method sustains with lesser training data as well as low computations, thereby making it more useful for scenarios with small training sets, restricted compute resources, or scenarios demanding a high degree of model understanding and stability during training.

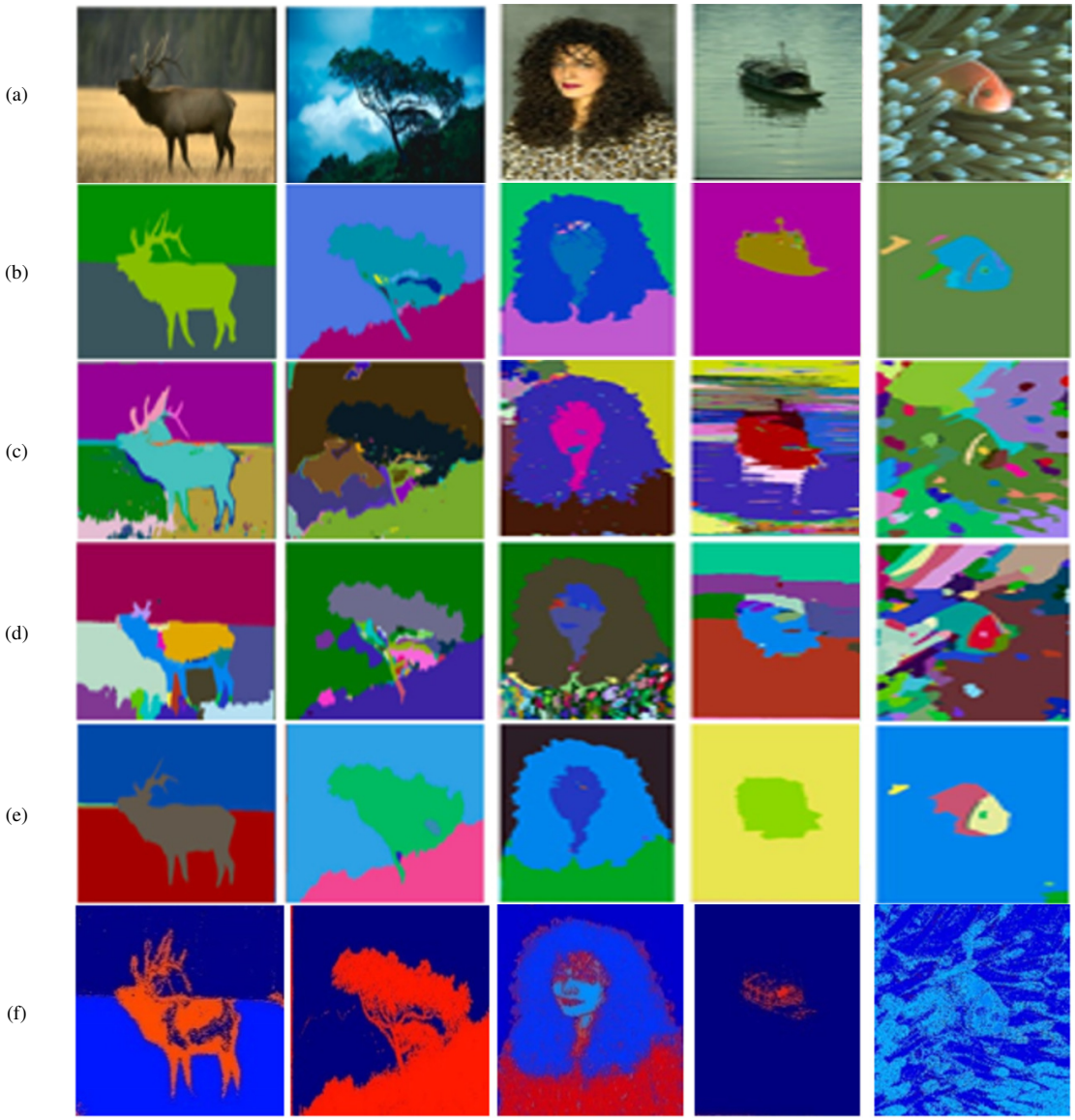

Fig. 10. Qualitative results obtained from the BSDS500 dataset: (a) original image, (b) ground truth, (c) FH [49], (d) DIC [50], (e) unsupervised image segmentation [51], and (f) proposed method.

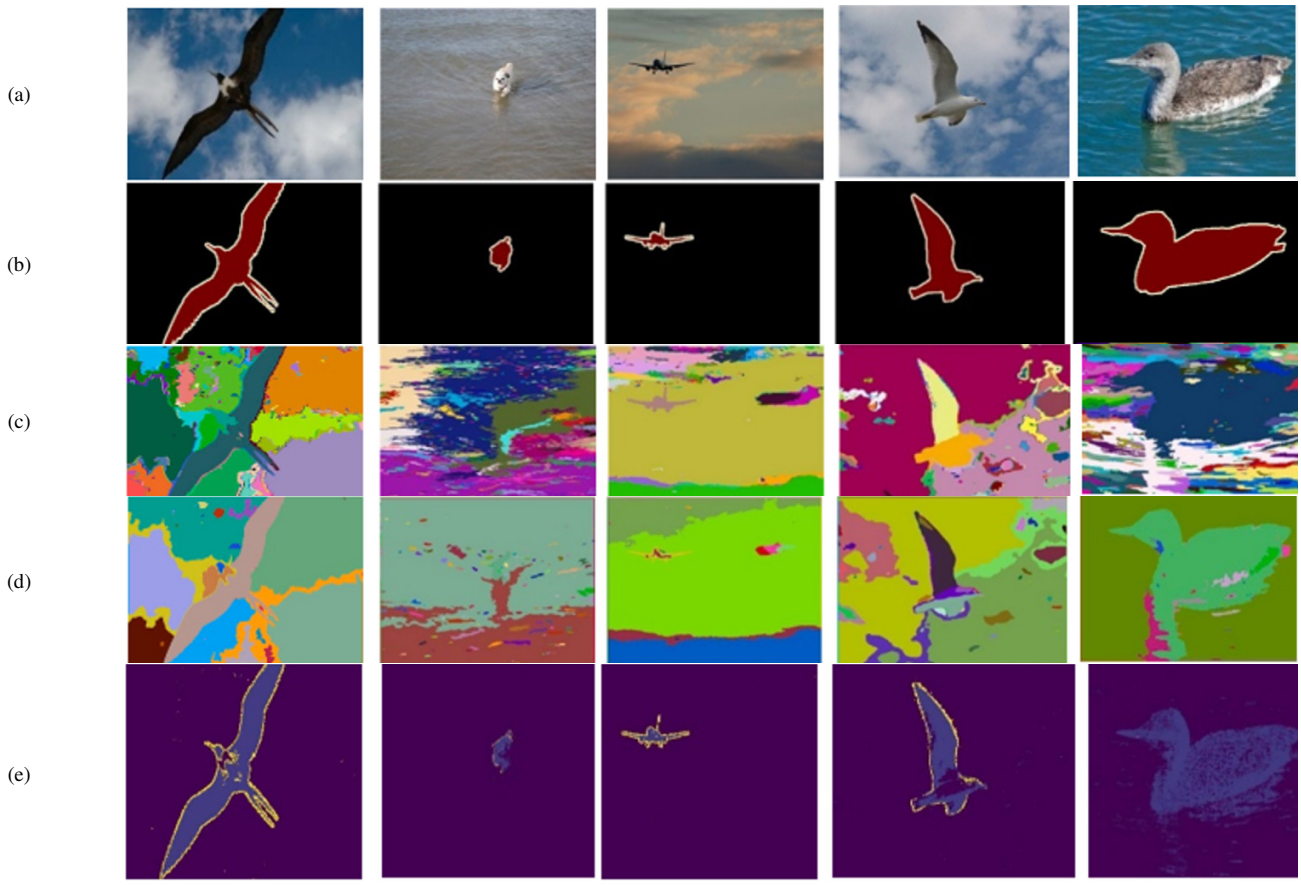

Fig. 11. Qualitative results on the PASCAL VOC 2012 dataset: (a) input image, (b) ground truth, (c) results from FH, DIC, (d) and (e) results obtained with the proposed method.

The comparison of qualitative results across both benchmark datasets demonstrates that the proposed framework achieves superior segmentation performance, consistently surpassing existing techniques. As illustrated in Figure 10, the outcomes on the BSDS500 dataset highlight the robustness of the approach. Each row displays, from top to bottom, the input image, the corresponding ground truth, and the segmentation outputs of FH, DIC, and the proposed method. Unlike conventional techniques that tend to either under-segment or excessively fragment regions, the proposed framework effectively partitions images into a balanced number of coherent clusters. This balance ensures both precision in boundary localization and preservation of structural details, thereby enhancing the interpretability of the segmentation results.

Figure 11 presents the qualitative analysis conducted on the PASCAL VOC 2012 dataset, further underscoring the generalizability of the method across diverse and complex scenes. In each row, the input image is followed by the ground truth annotation and the results generated by FH, DFC, and the proposed framework. The outputs reveal that the proposed method not only provides closer alignment with the ground truth but also demonstrates resilience against the challenges posed by cluttered backgrounds and varying object scales. In contrast, baseline methods frequently produce results with blurred boundaries or redundant segmentations, reducing their effectiveness for downstream tasks. Overall, the findings from both datasets establish that the proposed methodology excels in generating semantically meaningful and visually coherent segmentations. The qualitative superiority observed across multiple experimental settings demonstrates the reliability and adaptability of the approach, positioning it as a strong candidate for practical deployment in real-world applications.

V. CONCLUSION

This study presented an efficient and direct ensemble framework for image segmentation by integrating five advanced Machine Learning (ML)–based segmentation models using Weighted Majority Voting Ensemble (WMVE). The proposed system works in three main stages. In the preprocessing stage, Gaussian smoothing and median filtering were applied to enhance image quality and ensure reliable feature extraction. In the second stage, individual classifiers, Random Forest (RF), K-Nearest Neighbors (KNN), Naïve Bayes (NB), eXtreme Gradient Boosting (XGB), and Multilayer Perceptron (MLP), were trained using the prepared dataset. The third stage involved determining the weights of the classifiers based on their validation performance and aggregating their outputs through weighted voting to obtain the final segmentation results. The ensemble decision-making

process leveraged the strengths of individual classifiers, where each model contributed proportionally to its assigned weight.

This fusion approach improved segmentation accuracy and robustness by prioritizing the most reliable classifiers. To assess performance, several evaluation metrics, including Peak Signal-to-Noise Ratio (PSNR), Feature Similarity Index (FSIM), Segmentation Covering (SC), Probabilistic Rand Index (PRI), Variation of Information (VoI), Global Consistency Error (GCE), and Boundary Displacement Error (BDE), were employed. The experimental results demonstrated that the ensemble technique significantly enhanced the accuracy of individual models and consistently outperformed state-of-the-art ensemble and deep clustering methods. The proposed WMVE framework effectively achieved optimal segmentation without over-segmentation, producing near-perfect results across multiple datasets.

Furthermore, the study reviewed related ensemble learning research across various domains, categorizing prior work into traditional and deep learning–based ensembles. While the present work focused on specific datasets, it is anticipated that the proposed framework could generalize to broader contexts. However, the study has some limitations, since computational resource constraints in the Google Colab environment may limit scalability to larger datasets and more complex models. In addition, reliance on specific evaluation metrics under a 10-fold Cross-Validation (CV) framework assumes their adequacy in representing true model performance. Future work will extend the framework to larger datasets, optimize computational efficiency, and explore its applicability to additional segmentation domains.

DECLARATION OF COMPETING INTERESTS

The authors declare no competing interests.

ACKNOWLEDGMENTS

This study was supported by a research grant provided by the Research, Development, and Innovation Authority (RDIA), Kingdom of Saudi Arabia, with Grant No. 13382-psu-2023-PSNU-R-3-1-EI-. The authors would like to acknowledge the support of Prince Sultan University, Riyadh, Saudi Arabia, in paying the article processing charges of this publication. In addition, this research is supported by the Automated Systems and Computing Lab (ASCL), Prince Sultan University, Riyadh, Saudi Arabia.

DATA AVAILABILITY

The datasets used in this study: COCO 2014 [31], BSDS500 [32], and PASCAL VOC 2012 [33], are publicly available from their sources.

REFERENCES

[1] Z. Faska, L. Khrissi, K. Haddouch, and N. El Akkad, "A Robust and Consistent Stack Generalized Ensemble-Learning Framework for Image Segmentation," *Journal of Engineering and Applied Science*, vol. 70, no. 1, Dec. 2023, Art. no. 74, https://doi.org/10.1186/s44147-023-00226-4.

[2] Z. Faska *et al.*, "A Coherent Approach-Based Fine-Tuning of Segment Anything Model Plus Watershed Algorithm for Instance Segmentation of Mitochondria in Electron Microscopy Images," *IEEE Access*, vol. 13, pp. 98088–98105, 2025, https://doi.org/10.1109/ACCESS.2025.3574555.

[3] H. Moussaoui, N. El Akkad, and M. Benslimane, "A Brain Tumor Segmentation and Detection Technique Based on Birch and Marker Watershed," *SN Computer Science*, vol. 4, no. 4, Apr. 2023, Art. no. 339, https://doi.org/10.1007/s42979-023-01802-4.

[4] H. Moussaoui *et al.*, "Enhancing Automated Vehicle Identification by Integrating YOLO v8 and OCR Techniques for High-Precision License Plate Detection and Recognition," *Scientific Reports*, vol. 14, no. 1, Jun. 2024, Art. no. 14389, https://doi.org/10.1038/s41598-024-65272-1.

[5] M. A. Al-Ebrahim, "Spike-Based Attention Mechanisms for Enhanced Medical Image Segmentation," *Engineering, Technology & Applied Science Research*, vol. 15, no. 5, pp. 28273–28285, Oct. 2025, https://doi.org/10.48084/etasr.13407.

[6] L. Rokach, "Ensemble-Based Classifiers," *Artificial Intelligence Review*, vol. 33, no. 1–2, pp. 1–39, Feb. 2010, https://doi.org/10.1007/s10462-009-9124-7.

[7] A. Onan, S. Korukoğlu, and H. Bulut, "A Multiobjective Weighted Voting Ensemble Classifier Based on Differential Evolution Algorithm for Text Sentiment Classification," *Expert Systems with Applications*, vol. 62, pp. 1–16, Nov. 2016, https://doi.org/10.1016/j.eswa.2016.06.005.

[8] N. Sultana and M. M. Islam, "Meta Classifier-Based Ensemble Learning for Sentiment Classification," in *Proceedings of International Joint Conference on Computational Intelligence*, M. S. Uddin and J. C. Bansal, Eds. Singapore: Springer Nature Singapore, 2020, pp. 73–84.

[9] S. Karlos, G. Kostopoulos, and S. Kotsiantis, "A Soft-Voting Ensemble Based Co-Training Scheme Using Static Selection for Binary Classification Problems," *Algorithms*, vol. 13, no. 1, Jan. 2020, Art. no. 26, https://doi.org/10.3390/a13010026.

[10] Y. Zhang, H. Zhang, J. Cai, and B. Yang, "A Weighted Voting Classifier Based on Differential Evolution," *Abstract and Applied Analysis*, vol. 2014, pp. 1–6, 2014, https://doi.org/10.1155/2014/376950.

[11] B. Seijo-Pardo, I. Porto-Díaz, V. Bolón-Canedo, and A. Alonso-Betanzos, "Ensemble Feature Selection: Homogeneous and Heterogeneous Approaches," *Knowledge-Based Systems*, vol. 118, pp. 124–139, Feb. 2017, https://doi.org/10.1016/j.knosys.2016.11.017.

[12] H. Kim, H. Kim, H. Moon, and H. Ahn, "A Weight-Adjusted Voting Algorithm for Ensembles of Classifiers," *Journal of the Korean Statistical Society*, vol. 40, no. 4, pp. 437–449, Dec. 2011, https://doi.org/10.1016/j.jkss.2011.03.002.

[13] S. K. Satapathy, A. K. Jagadev, and S. Dehuri, "Weighted Majority Voting Based Ensemble of Classifiers Using Different Machine Learning Techniques for Classification of EEG Signal to Detect Epileptic Seizure," *Informatica*, vol. 41, no. 1, Mar. 2017.

[14] L. Breiman, "Stacked Regressions," *Machine Learning*, vol. 24, no. 1, pp. 49–64, Jul. 1996, https://doi.org/10.1023/A:1018046112532.

[15] S. K. Warfield, K. H. Zou, and W. M. Wells, "Simultaneous Truth and Performance Level Estimation (STAPLE): An Algorithm for the Validation of Image Segmentation," *IEEE Transactions on Medical Imaging*, vol. 23, no. 7, pp. 903–921, Jul. 2004, https://doi.org/10.1109/TMI.2004.828354.

[16] T. Rohlfing and C. R. Maurer, "Shape-Based Averaging for Combination of Multiple Segmentations," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2005*, vol. 3750, J. S. Duncan and G. Gerig, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 838–845.

[17] L. K. Hansen and P. Salamon, "Neural Network Ensembles," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 10, pp. 993–1001, Oct. 1990, https://doi.org/10.1109/34.58871.

[18] V. Singh, L. Mukherjee, J. Peng, and J. Xu, "Ensemble Clustering Using Semidefinite Programming with Applications," *Machine Learning*, vol. 79, no. 1–2, pp. 177–200, May 2010, https://doi.org/10.1007/s10994-009-5158-y.

[19] Y. Zuo and T. Drummond, "Fast Residual Forests: Rapid Ensemble Learning for Semantic Segmentation," in *Proceedings of the 1st Annual Conference on Robot Learning*, Mountain View, CA, USA, 2017, vol. 78.

[20] A. Holliday, M. Barekatain, J. Laurmaa, C. Kandaswamy, and H. Prendinger, "Speedup of Deep Learning Ensembles for Semantic

Segmentation using a Model Compression Technique," *Computer Vision and Image Understanding*, vol. 164, pp. 16–26, Nov. 2017, https://doi.org/10.1016/j.cviu.2017.05.004.

[21] I. Nigam, C. Huang, and D. Ramanan, "Ensemble Knowledge Transfer for Semantic Segmentation," in *2018 IEEE Winter Conference on Applications of Computer Vision*, Lake Tahoe, NV, USA, Mar. 2018, pp. 1499–1508, https://doi.org/10.1109/WACV.2018.00168.

[22] D. Marmanis, J. D. Wegner, S. Galliani, K. Schindler, M. Datcu, and U. Stilla, "Semantic Segmentation of Aerial Images with an Ensemble of CNNs," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. III–3, pp. 473–480, Jun. 2016, https://doi.org/10.5194/isprs-annals-III-3-473-2016.

[23] Y. Koren, The BellKor solution to the Netflix Grand Prize, Haifa, Israel: Yahoo, 2009.

[24] Y. W. Kim, J. Innila Rose, and A. V. N. Krishna, "Accuracy Enhancement of Portrait Segmentation by Ensembling Deep Learning Models," in *2020 Fifth International Conference on Research in Computational Intelligence and Communication Networks*, Bangalore, India, Nov. 2020, pp. 59–64, https://doi.org/10.1109/ICRCICN50933.2020.9296196.

[25] T. Cover and P. Hart, "Nearest Neighbor Pattern Classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, Jan. 1967, https://doi.org/10.1109/TIT.1967.1053964.

[26] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001, https://doi.org/10.1023/A:1010933404324.

[27] T. G. Dietterich, "Ensemble Methods in Machine Learning," in *Multiple Classifier Systems*, vol. 1857, Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 1–15.

[28] N. Littlestone and M. K. Warmuth, "The Weighted Majority Algorithm," *Information and Computation*, vol. 108, no. 2, pp. 212–261, Feb. 1994, https://doi.org/10.1006/inco.1994.1009.

[29] Z. B. Zabinsky, "Random Search Algorithms," in *Wiley Encyclopedia of Operations Research and Management Science*, 1st ed., Hoboken, NJ, USAWiley: 2011.

[30] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A Database of Human Segmented Natural Images and Its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics," in *Proceedings Eighth IEEE International Conference on Computer Vision*, Vancouver, BC, Canada, 2001, vol. 2, pp. 416–423, https://doi.org/10.1109/ICCV.2001.937655.

[31] T.-Y. Lin *et al.*, "Microsoft COCO: Common Objects in Context," in *Computer Vision – ECCV 2014*, vol. 8693, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 740–755.

[32] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour Detection and Hierarchical Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 898–916, May 2011, https://doi.org/10.1109/TPAMI.2010.161.

[33] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes Challenge: A Retrospective," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, Jan. 2015, https://doi.org/10.1007/s11263-014-0733-5.

[34] D. Reynolds, "Gaussian Mixture Models," in *Encyclopedia of Biometrics*, S. Z. Li and A. Jain, Eds. Boston, MA: Springer US, 2009, pp. 659–663.

[35] W. Kim, A. Kanezaki, and M. Tanaka, "Unsupervised Learning of Image Segmentation Based on Differentiable Feature Clustering," *IEEE Transactions on Image Processing*, vol. 29, pp. 8055–8068, 2020, https://doi.org/10.1109/TIP.2020.3011269.

[36] X. Xia and B. Kulis, "W-Net: A Deep Model for Fully Unsupervised Image Segmentation." arXiv, 2017, https://doi.org/10.48550/ARXIV.1711.08506.

[37] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep Clustering for Unsupervised Learning of Visual Features," in *Computer Vision – ECCV 2018*, vol. 11218, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 139–156.

[38] Y. Pu, J. Sun, N. Tang, and Z. Xu, "Deep Expectation-Maximization Network for Unsupervised Image Segmentation and Clustering," *Image and Vision Computing*, vol. 135, Jul. 2023, Art. no. 104717, https://doi.org/10.1016/j.imavis.2023.104717.

[39] A. Kanezaki, "Unsupervised Image Segmentation by Backpropagation," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing*, Calgary, AB, Canada, Apr. 2018, pp. 1543–1547, https://doi.org/10.1109/ICASSP.2018.8462533.

[40] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, Aug. 2000, https://doi.org/10.1109/34.868688.

[41] C. Wang, B. Yang, and Y. Liao, "Unsupervised Image Segmentation Using Convolutional Autoencoder with Total Variation Regularization as Preprocessing," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing*, New Orleans, LA, USA, Mar. 2017, pp. 1877–1881, https://doi.org/10.1109/ICASSP.2017.7952482.

[42] D. Comaniciu and P. Meer, "Mean Shift: A Robust Approach Toward Feature Space Analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, May 2002, https://doi.org/10.1109/34.1000236.

[43] S. Xie and Z. Tu, "Holistically-Nested Edge Detection." arXiv, 2015, https://doi.org/10.48550/ARXIV.1504.06375.

[44] Y. Liu *et al.*, "Richer Convolutional Features for Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 8, pp. 1939–1946, Aug. 2019, https://doi.org/10.1109/TPAMI.2018.2878849.

[45] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, Apr. 2018, https://doi.org/10.1109/TPAMI.2017.2699184.

[46] J. He, S. Zhang, M. Yang, Y. Shan, and T. Huang, "Bi-Directional Cascade Network for Perceptual Edge Detection." arXiv, 2019, https://doi.org/10.48550/ARXIV.1902.10903.

[47] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, vol. 9351, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham: Springer International Publishing, 2015, pp. 234–241.

[48] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017, https://doi.org/10.1109/TPAMI.2016.2644615.

[49] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient Graph-Based Image Segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, Sep. 2004, https://doi.org/10.1023/B:VISI.0000022288.19776.77.

[50] L. Zhou and W. Wei, "DIC: Deep Image Clustering for Unsupervised Image Segmentation," *IEEE Access*, vol. 8, pp. 34481–34491, 2020, https://doi.org/10.1109/ACCESS.2020.2974496.

[51] C. M. Hoang and B. Kang, "Pixel-Level Clustering Network for Unsupervised Image Segmentation," *Engineering Applications of Artificial Intelligence*, vol. 127, Jan. 2024, Art. no. 107327, https://doi.org/10.1016/j.engappai.2023.107327.