https://doi.org/10.1186/s44147-023-00226-4

Journal of Engineering and Applied Science

RESEARCH **Open Access**

A robust and consistent stack generalized ensemble-learning framework for image segmentation

Zahra Faska[1*], Lahbib Khrissi[2], Khalid Haddouch[1] and Nabil El Akkad[1]

*Correspondence:
zahra.faska@usmba.ac.ma

[1] Laboratory of Engineering, Systems and Applications, ENSA, Sidi Mohamed Ben Abdellah University, Fez, Morocco
[2] LIIAN, Department of Computer Science, FSDM of Fez, Sidi Mohamed Ben Abdellah University, Fez, Morocco

Abstract

In the present study, we aim to propose an effective and robust ensemble-learning approach with stacked generalization for image segmentation. Initially, the input images are processed for feature extraction and edge detection using the Gabor filter and the Canny algorithms, respectively; our main goal is to determine the most feature descriptions. Subsequently, we applied the stacking generalization technique, which is generally built with two main learning levels. The first level is composed of two algorithms that give good results in the literature, namely: LightGBM (Light Gradient Boosting Machine) and SVM (support vector machine). The second level is the meta-model in which we use a predictor model that takes the base-level predictions to improve the accuracy of the final prediction. In the stacked generalization process, we use the Extreme Gradient Boosting (XGBoost); it takes as input the sub-models' outputs to better classify each pixel of the image to give the final prediction. Today, several research works exist in the literature using different machine learning algorithms; in fact, instead of trying to find a single efficient and optimal learner, ensemble-based techniques take the advantage of each basic model; they integrate their outputs to obtain a more consistent and reliable learner. The result obtained from the models of individuals and our proposed approach is compared using a set of evaluation measures for image quality such as IoU, DSC, CC, SSIM, SAM, and UQI. The evaluation and a comparison of the results obtained showed more consistent predictions for the proposed model. Thus, we have made a comparison with some recent deep learning-based unsupervised segmentation methods. The evaluation and a comparison of the results obtained showed more coherent predictions for our stacked generalization in terms of precision, robustness, and consistency.

Keywords: Image segmentation, Edge detection, Ensemble learning, XGBoost, LightGBM, SVM, Stacking

Introduction

The image segmentation is considered the most critical function and the most important process of image processing and analysis. The goal of image segmentation is to divide or partition a digital image into regions (set of pixels) that are homogeneous and inhomogeneous according to some criteria. All pixels in a region are similarly based on some image characteristics, namely: the color, intensity value, and texture. There are many applications of image segmentation in the literature such as camera self-calibration, 3D reconstruction, medical imaging, and cryptography. Image segmentation is considered the most critical function and the most important process of image processing and analysis. The goal of image segmentation is to divide or partition a digital image into regions (set of pixels) that are homogeneous and inhomogeneous according to some criteria. All pixels in a region are similarly based on some image characteristics, namely: the color, intensity value, and texture. There are many applications of image segmentation in the literature such as camera self-calibration, 3D reconstruction, medical imaging, and cryptography. Image segmentation is considered the most important and most difficult process of image processing and analysis because of several constraints (the influence of complicated background, variety of characteristics of the object, and noise). Currently, a rich amount of literature on image segmentation has been published over the past decades, but each method proposed is valid just for a given type of image in a given computer context. There are many image segmentation techniques including clustering [1, 2], split/merge [3], region growth [4], active contour [5], SVM [6], random forest [7], genetic algorithms [8], and CNN [9]. However, image segmentation techniques are grouped into five techniques [10]. The first technique is segmentation by edge detection approach; this method consists of finding boundaries separating regions when there is a sudden change in intensity value or else regions of different textures. This approach can be classified into three categories of methods: the first- or second-order derivatives method, deformable methods, and analytical methods [11, 12]. The second technique is the segmentation by region; this category aims to segment the image into various regions having similar characteristics, where we generally have region growing and split-and-merge algorithms [13]. The third technique is threshold-based segmentation [14]. This approach is widely used to detect different objects in the image by using threshold values based on classification rules. When we need only one object in an image, the rest of the image is called the background. These methods divide the image pixels concerning their intensity level. However, the challenge in this is to find an appropriate threshold. In the fourth category, we have watershed-based segmentation [15]. This method uses the concept of topological interpretation, where the gradient of the image is considered a topographic surface, and the intensity value represents the height. The minimum value of this height is assigned to a region and the maximal one to the edge. The pixels with more gradient are represented as boundaries. However, the generation of noise remains a problem in front of the direct application of this method, which can lead to the problem of overfitting. The fifth technique is segmentation by clustering [16]. This method tries to segment the image into clusters having pixels with similar characteristics.

Image segmentation is an important and difficult research issue on image processing. To cope with shortcomings of segmentation algorithms that been proposed have affirmed their limits, and to answer the question "how good is a given segmentation algorithm?", the researcher's ingenuity led them to propose performance measurements and to explore other potentially effective tools and search new, more efficient and powerful techniques for good segmentation.

This article aims to develop and test a stacked generalization framework based on an ensemble-learning approach containing two basic models followed by a meta-learner.

The meta-learner takes predictions of sub-models as input and learns how to best combine them to make a better output prediction. To verify that the ensemble model successfully integrated the outputs of the sub-models, we compared it with the individual models to show that the stacking generalization approach we have proposed can give a better result for image segmentation.

The organization of the sections of our work is as follows: the "Brief literature review" section is brief literature on some of the essential concepts for this paper including the ensemble-learning algorithm, XGBoost, LightGBM, and SVM. The "Methods" section provides the theoretical foundation for our framework. In the "Results and discussion" section, we present the schematic diagram and proposed framework structure. The "Results and discussion' section will be consecrated for experiments and comparison of results. Discussions and conclusions will be addressed in the "Results and discussion" sections.

Brief literature review

Introduction

The image segmentation is a very broad research axis; we found today several research works are published in the literature using different machine learning algorithms, but we can notice that all the proposed methods have affirmed their limits. So it becomes necessary to find other more flexible and reliable methods. Instead of choosing the best algorithm to do the segmentation, a stacking ensemble technique gave us a more robust classifier because it combines the output of a set of base models rather than trying to provide a single optimal learner.

XGBoost ("extreme gradient boosting") was proposed by Chen and Guestrin [17]. More recently, it has been very successful and has attracted wide attention because of its high efficiency and high prediction accuracy. XGBoost is an optimized GBDT ("gradient boosting decision tree") algorithm, which consists of many decision trees. The GBDT is proven by Yang, Wang, and Zhang [18]; Zhao, Zheng, and Li [19]; and Wang, Deng, and Wang [20]. However, XGBoost is more efficient compared with other machine-learning algorithms; among them are SVM, decision tree (DT), and GBDT. During the XGBoost modeling process, each decision tree (DT) depends on the result of the previous tree to provide a more powerful predictor [21]. This modeling process is generally very fast [22]. In addition, the term regularization is integrated with this process to avoid the problem of overfitting and reduce the complexity of the model. XGBoost belongs to the DMLC ("distributed machine learning community"). Its library is designed to be efficient, flexible, and portable [23]. On the other hand, XGBoost also optimizes memory resources and manages missing values during the learning process (sparse aware [24]).

While the algorithm is a scalable and efficient tree boosting system, which is generally used in the field of classification and regression [25], during classification problems, XGBoost presents weak and less accurate results with unbalanced data (when one or more classes have lower proportions in a dataset than the other classes [26]).

Furthermore, LightGBM is a newly developed technique. It was designed by Microsoft Research Asia [27]. It is another innovative machine-learning algorithm with its remarkable proficiency, accuracy in data classification, and regression with a very short accuracy time. LightGBM develops trees with the principle of leaf-wise split approach instead of level-wise approach. It searches for maximum profit nodes during the division process. Therefore, in cases where memory consumption, processing time, and arithmetic speed are considered, the LightGBM becomes an excellent choice for faster training, adequate efficiency, optimal memory, computer utilization satisfactory accuracy, parallelism, and large-scale data processing capabilities. The downside is that the information in the discarded leaves may be ignored, which makes the split results insufficiently detailed.

SVM is a supervised machine-learning algorithm, developed by Vapnik and Cortes [28]. This method is based on the idea of finding a hyperplane that linearly separates feature vectors in high-dimensional spaces. Good generalization ability could ensure higher classification accuracy when there are fewer training samples by minimizing the Vapnik-Chervonenkis (VC) dimension and achieving minimal structural risk [29]. In fact, SVM is very popular due to its speed, generation capacity, no restrictive data assumptions, and flexibility (prior knowledge can be used to tune its kernels in an easy way [30, 31]). On the other hand, when we have high-dimensional data (the distribution of the data in the high-dimensional feature space is different from the input space), this method may not be optimal.

Since the individual algorithms have asserted their limits and their shortcomings, it becomes necessary to propose and explore other potentially efficient and powerful tools. In this axis, research has thought of combining the advantages of different models to overcome the weak points and problems mentioned above [32]. Ensemble-learning is based on the idea of increasing the generalization performance of the model by using several machine learning tools and pooling them to obtain better prediction results. The ensemble-learning method assumes that the performance of each expert is measurable to construct the final decision [33] in order to obtain more precise and more stable results [34]. Ensemble learning uses some ensemble strategies like voting, averaging, and learning [35, 36]. However, the stacking learning method is also an ensemble method that is used to obtain results with better output prediction. In general, the stack consists of two main layers: the first level is called “the base model” (of more than two models), and the second level is “the meta-model.” This last level combines the base model outputs by integrating the advantages of the different models; with the stacking method, one can correct the errors in the base model to improve the integrated model accuracy.

Motivated by the advantages of the stacking ensemble technique, this research developed a stacking ensemble technique for image segmentation, taking the integration of two models (SVM and LightGBM) as the input to the meta-model, which is XGBoost in our case.

Methods

Extreme Gradient Boosting (XGBoost)

GBDT is an ensemble ML algorithm using multiple DTs as base learners. Every decision tree (DT) is not independent, because a new added DT increases emphasis on the misclassified samples attained by previous DTs [37]. The diagram of GBDT algorithm is shown in Fig. 1. It can be noticed that the residual of former DTs is taken as the input for the next DT. Then, the added DT is used to reduce residual, so that the

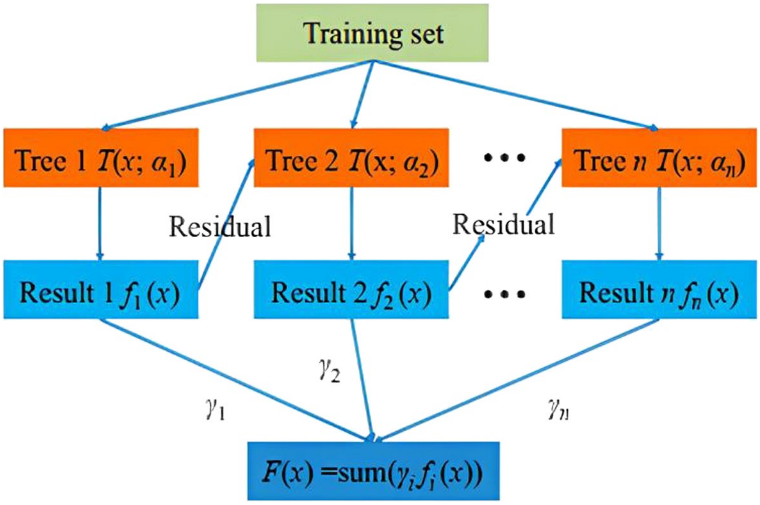


Fig. 1 Diagram of GBDT algorithm

loss decreases following the negative gradient direction in each iteration. Finally, the prediction result is determined based on the sum of results from all DTs.

XGBoost is a very popular new ML model. It is based on the structure of GBDT and is used in many fields because it is considered a reliable and efficient solution to several machine-learning problems [38]. It has been triumphant in many machine learning competitions like Kaggle [39]. In the modeling process of this algorithm, the regularization term is integrated to control overfitting, which gives it better performance. Additionally, XGBoost provides an improved classifier through a set of weak classifiers. In fact, XGBoost has known a great success compared to other gradient boosting algorithms, thanks to its high flexibility, and speed, support regularization, enabled cross-validation, and is designed to handle missing data with its in-build features. XGBoost is essentially used to minimize the loss function with the addition of weak classifiers, with other terms to minimize the regularized objective as follows:

$$Obj(\theta) = \sum_i L(\widehat{Y}_i, Y_i) + \sum_k \Omega(f_k) \tag{1}$$

where $\Omega\left(f_k\right) = \gamma T + \frac{1}{2}\lambda\|\omega\|^2$

Here, l denotes the loss function that measured the difference between the prediction $\widehat{Y_i}$ and the target Y_i. The Ω (.) penalized the complexity of the model (i.e., the regression tree functions). The additional regularization term helped to smooth the final learnt weights to avoid over-fitting. In addition, XGBoost uses a set of parameters to find an optimal tree structure in order to minimize the objective function.

For each training case and each boosting iteration for the objective function "squared error," the first- and second-order gradient was calculated in XGBoost. The model was built using the XGBoost library, which is compatible with scikit-learn. Figure 2 represents the XGBoost regression mechanism.

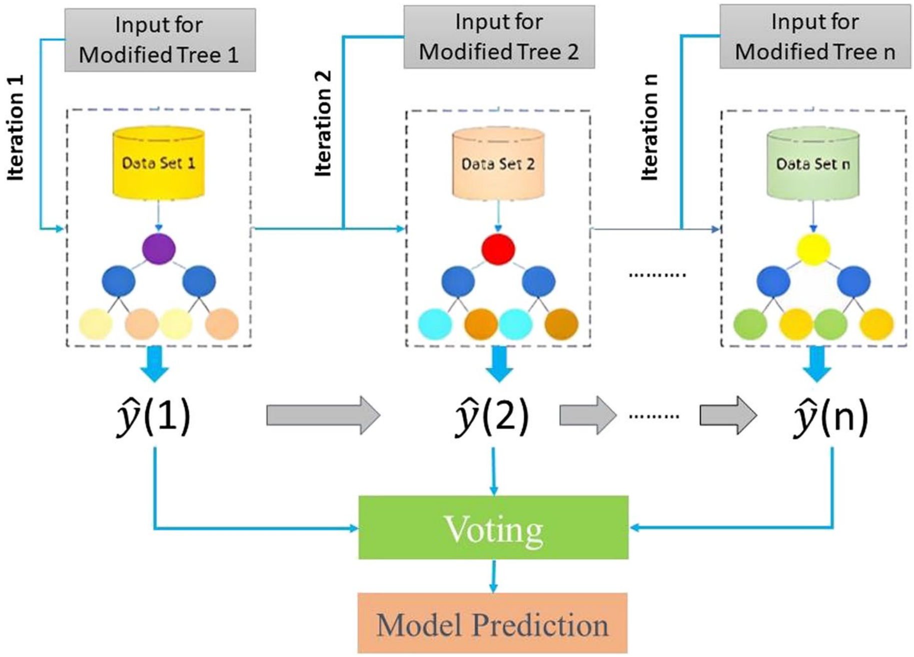


Fig. 2 Extreme Gradient Boosting Machine (XGBoost) regression

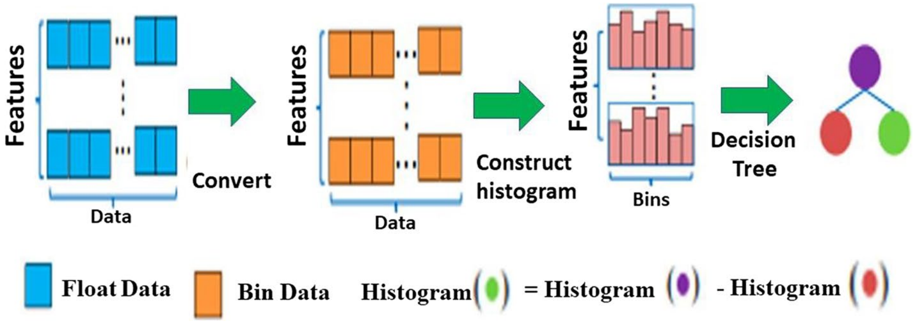


Fig. 3 Histogram-based decision tree algorithm

Light Gradient Boosting Machine (LGBM)

Light Gradient Boosting Machine (LightGBM) is another innovative gradient boosting framework, which was developed by Microsoft MSRA in 2016 by combining two new techniques: EFB (exclusive feature bundling) and GOSS (Gradient-based One-Side Sampling) [40]. LightGBM has achieved considerable success on regression and classification problems and other machine learning tasks with a relatively short processing time. LightGBM offered to solve the problem faced by GBDT regarding larger data. The objective is to make GBDTs better used with a very fast training time. LightGBM selects histogram-based decision tree algorithm and splits nodes by splitting cells with tree depth control and minimum data of each node to avoid fitting problem.

Firstly, LightGBM creates a histogram as Fig. 3 shows. This histogram classifies continuous feature values into discrete groups, constructed using a subset of the dataset. Since the histogram is based on discrete values instead of sorted values, one can find an optimal segmentation point [41]. This method is more efficient in terms of both memory consumption and speed.

Secondly, LightGBM uses leaf-wise instead of the traditional decision tree splitting strategy, which is level-wise. Actually, the two strategies are different as is shown in Fig. 4. Leaf-wise enlarges the tree looking for nodes of maximum loss change during the splitting process. On the other hand, level-wise divides each node at each level, and consequently, this requires large memory resources and high computation costs. Level-wise growth is usually better for smaller datasets whereas leaf-wise tends to overfit. Leaf-wise growth tends to excel in larger datasets where it is considerably faster and more efficient than level-wise growth.

However, readers who want to have a deeper understanding of LightGBM algorithms can refer to the references made by Guolin Ke et al. [37], where the principles and applications of LightGBM algorithms are described in detail.

Support vector machines (SVMs)

SVM is a family of supervised machine learning algorithms and can be used for classification or regression problems. SVMs are a class of algorithms based on the “structural risk” minimization principle described by statistical learning theory that uses linear separation. This consists of finding the optimal hyperplane limit that better separates the training data in order to make a better distinction between the models. This limit can be defined through different kernels [42]. However, Cortes [43] presents a more in-depth mathematical explanation of this algorithm.

Methods

The proposed stack generalized machine learning architecture used on this paper is shown in Fig. 5. First, each model processes the input image independently. Then the meta-model takes as its input the output predictions of all these models; it tries to combine them and integrate their advantages to obtain a better output prediction.

The stacking technique is an ensemble-learning algorithm, initially proposed by Wolpert [44] and based on the “winner-takes-all” principle. In fact, instead of trying to find a

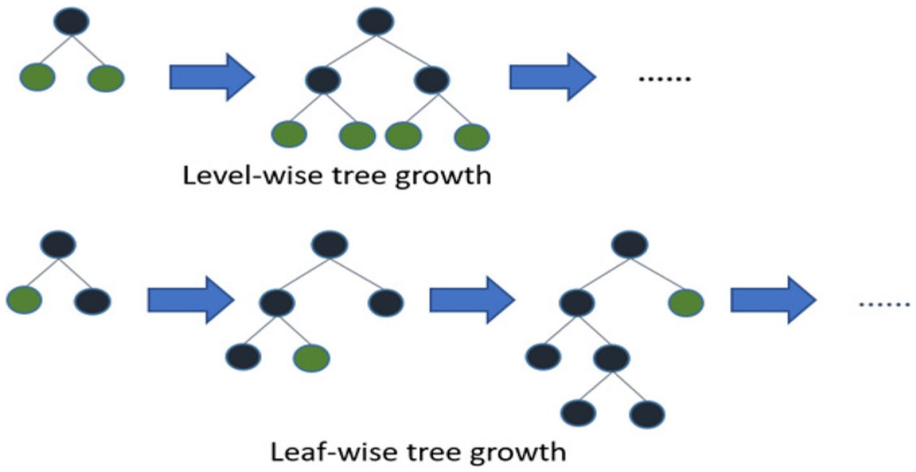


Fig. 4 Level-wise and leaf-wise tree construction

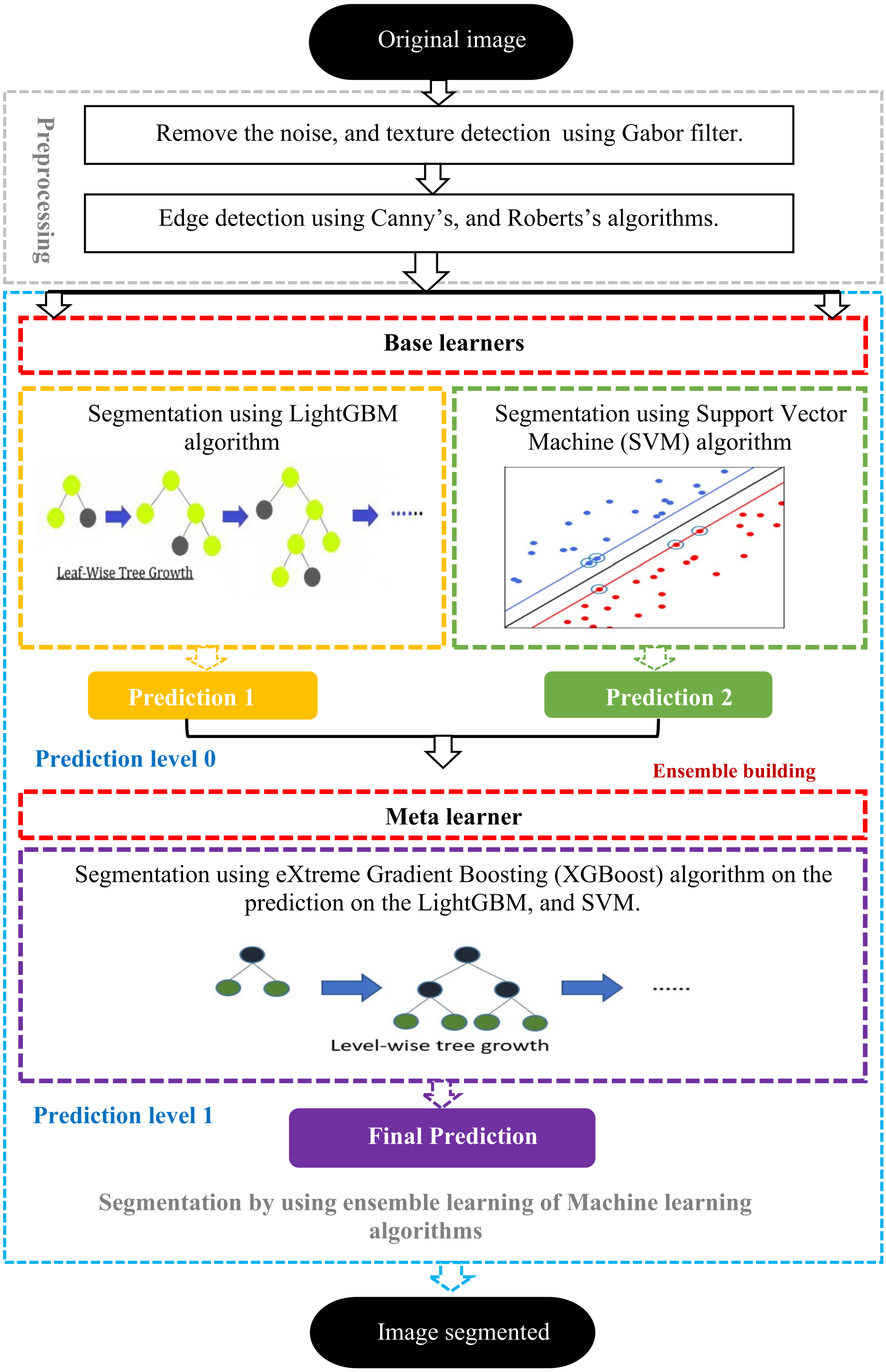

Original image
Preprocessing
Remove the noise, and texture detection using Gabor filter.
Edge detection using Canny's, and Roberts's algorithms.
Base learners
Segmentation using LightGBM algorithm
Leaf-Wise Tree Growth
Segmentation using Support Vector Machine (SVM) algorithm
Prediction 1
Prediction 2
Prediction level 0
Ensemble building
Meta learner
Segmentation using eXtreme Gradient Boosting (XGBoost) algorithm on the prediction on the LightGBM, and SVM.
Level-wise tree growth
Prediction level 1
Final Prediction
Segmentation by using ensemble learning of Machine learning algorithms
Image segmented

Fig. 5 The flow chart of the proposed method

single efficient and optimal learner, ensemble-based techniques, as the name suggests, take the advantage of each basic pattern; they integrate their outputs in order to obtain a more robust and reliable learner. In general, ensemble models could be utilized for both classification and regression [45]. The stacking generalization method [46] is part of the ensemble-learning family, in which another model takes predictions from a set of weak learners as its input and combines them to give improved prediction accuracy.

The overall learning pipeline is consisting of three stages:

Image processing

In the image processing stage, an ensemble of filters to produce texture features and reduction and edge detection are applied. The input for our proposed method is a color image. For optimal texture separability, we are using Gabor filter, and for edge detection, we have applied Canny and Robert's filter.

Gabor filter

The Gabor filter is a linear filter often used for edge extraction and texture features. Many researchers claim that the frequency and direction representations of the Gabor filter are close to those of human visual systems. It is considered one of the most popular texture segmenting methods, which obtained the response of the texture after filtering it through different orientations and then extracted textual features for segmentation. However, due to its flexibility in different orientations and frequencies, the Gabor filter has become a very useful tool for extracting and analyzing the texture features and for detecting the image edges.

The 2D Gabor filter consists of a sinusoidal plane wave and a Gaussian kernel in the spatial domain, which has the following mathematical expressions:

$$g\left(x, y^{'}, \lambda, \theta, \psi, \sigma, y\right) = \exp\left(\frac{x^{2^{'}} + y^{2}y^{2^{'}}}{2\sigma^{2}}\right)\exp(i\left(2\pi\frac{X^{'}}{\lambda} + \psi\right)) \tag{2}$$

where

$$X^{'} = X\cos\theta + Y\sin\theta$$

$$Y^{'} = -X\sin\theta + Y\cos\theta$$

where λ and θ respectively control the wavelength of the sinusoidal component and the orientation of the Gabor filters; ψ represents the phase shift; σ is the Gaussian standard deviation; and γ is the spatial aspect ratio.

Canny operator

The Canny method was first proposed by John Canny in 1986 [47]. The algorithm has been widely used in various computer vision and pattern recognition systems. This technique is very useful for extracting the edges of the image using the first and second derivatives of gray as a function of several characteristics, which are presented by the large change and discontinuity in the value of gray on the edge of the image. The Canny method has three clearly explained criteria for optimizing the edge detection:

1) Detection of the edges with a low error rate, with losing important edges or appearing false edges, maximizes the signal-to-noise ratio accurately.

2) The edges detected by the algorithm need to be located precisely in the center of the edge.
3) Only one response on a single contour means each edge in the image must be marked only once.

Base models' construction

In the construction stage of our framework, we developed two levels of classifiers. The first level is called "the base model"; it consists of a set of two different machine learning models: LightGBM and SVM. The choice of these two algoritms is based on their great success in the field of classification, on hand, SVM thanks to its generation capacity and its flexibility, and on the other hand, LightGBM with its proficiency and short time processing time. Moreover, the original image is transmitted to all these basic learners, who will be trained individually and separately in order to give us a prediction with a difference in terms of precision at the end of the execution of each algorithm. Then, this output obtained by the basic model will be exploited and transmitted for another segmentation process; this is the second level of classifiers called "the meta model."

Meta-model combination

In after receiving the base model predictions, the stacking technique is used in this step to get the combined output. In fact, the meta-model uses a predictor model, which has as input the base predictions and not the input data. Consequently, our meta-model is another classifier (XGBoost); its role is to integrate the advantages of the basic model and to try to better classify each pixel of the image to give the final prediction.

Results and discussion

In this part, we will present the experiments and the results obtained by our proposed approach in order to make a global evaluation and validate its robustness and efficiency in the field of image segmentation. The method used in our research has been tested on Berkeley Segmentation Dataset and Benchmark (BSD500) [48], not so large, and contains only 500 images with ground truth labels. To justify the results, we have also provided qualitative and quantitative comparisons of performances between our stacked generalization framework and other individual models giving good results in the literature, respectively: Light Gradient Boosting Machine (LightGBM), support vector machine (SVM), and Extreme Gradient Boosting (XGBoost), in image segmentation of buildings from the source of the same image.

Most of machine learning models have several important parameters that need to be tuned because they control the accuracy of the model. In the literature, there are several techniques used to calculate the optimal values of these hyperparameters; the widely used are as follows: grid search, Bayesian optimization, heuristic search, and randomized search [49]. In this proposed approach, some hyperparameters in SVM, XGBoost, and LightGBM algorithms are tuned using the grid search infrastructure in scikit-learn. Parameter values and meanings of these methods are presented in Table 1.

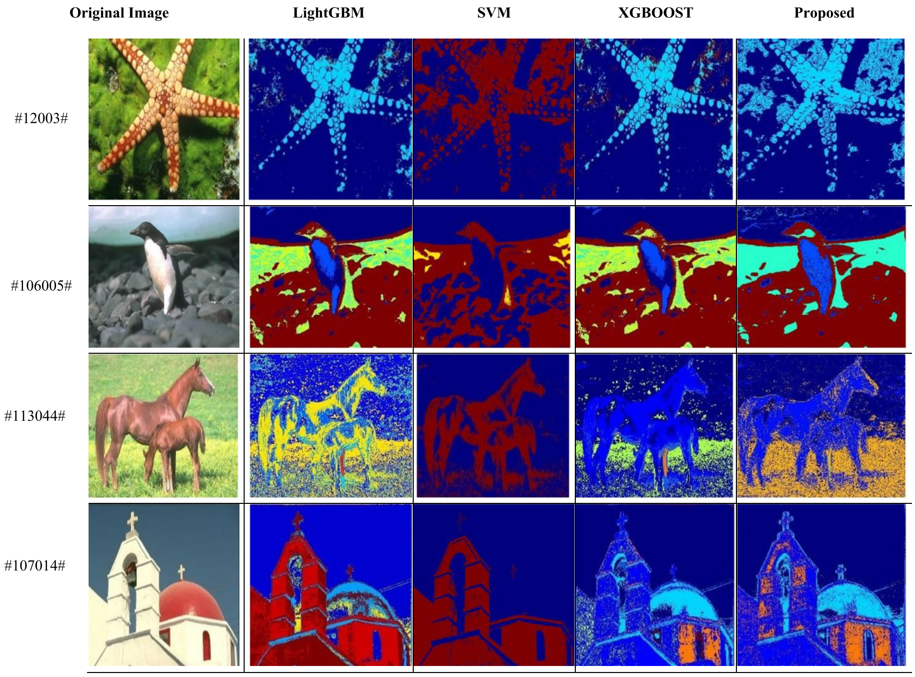


Fig. 6 Image segmentation results obtained by the different methods

The grid search technique is a tuning method that attempts to optimize the hyperparameter values of a model [50]. Its optimization process is as follows: first, the model is trained by running through different hyperparameter combinations of all possible values of each parameter. Each combination corresponds to a model by comparing the calculated error of the model to select the hyperparameters that can improve learning ability and prediction accuracy of the model.

In our experiments, LightGBM, SVM, and XGBoost are implemented using the scikit-learn, the XGBoost, and the LightGBM libraries in Python 3.7. The test and experiment were carried out on a Windows 10 64-bit laptop equipped with Intel Core™ i5-5200U CPU and 8G RAM.

Table 1 Hyperparameters of the algorithms and their values

Algorithm	Hyperparameters	Meanings	Value
XGBoost	n_estimators	Number of trees	1000
	learning_rate	Shrinkage coefficient of each tree	0.2
	max_depth	Maximum depth of a tree	50
	colsample_bytree	Subsample ratio of columns for tree construction	1
	subsample	Subsample ratio of training samples	0.3
LightGBM	n_estimators	Number of trees	1900
	learning_rate	Shrinkage coefficient of each tree	0.2
	max_depth	Maximum depth of a tree	80
	num_leaves	Number of leaves for each tree	11
SVM	max_iter	Hard limit on iterations within solver	100
	C	Regularization parameter	1.2

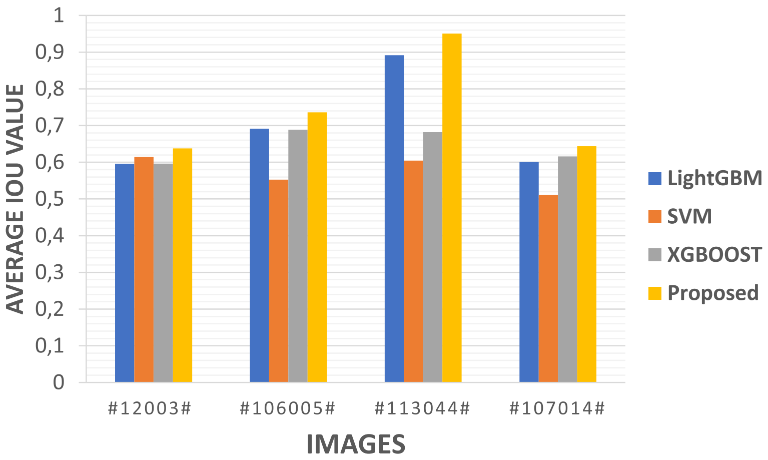


Fig. 7 IoU values obtained by individual's algorithms and our proposed technique

From the Fig. 6, we can notice that the proposed method in this contribution, which is based on stacking generalization, has succeeded in integrating basic learner predictions and then trying to combine them to obtain a more robust and optimal classifier for better segmentation.

In this study, the segmentation performances of the proposed approach are evaluated and the results are compared over a set of best quality measures such as IOU, DSC, CC, SSIM, SAM, and UQI. IOU and the Dice similarity coefficient (DSC) statistical parameter values are used to analyze the quality of the segmented image.

The Intersection over Union (IoU) also known as Jaccard index or Jaccard similarity coefficient is an evaluation metric used to calculate the performance of segmentation models. It is generally defined as the ratio of intersection and union area between the target mask and our prediction output. IOU is defined by the following:

$$IoU = \frac{|A \cap B|}{|A \cup B|} \tag{3}$$

where B and A represent the predicted segmentation maps and ground truth, respectively.

The Dice similarity coefficient (DSC), also called the Sorensen-Dice index or simply the Dice coefficient, is a statistical tool that measures the spatial overlap between two segmentations, A and B target regions, and is defined as follows:

$$DSC = \frac{2|A \cap B|}{|A| + |B|} \tag{4}$$

Note that higher IoU and DSC and value demonstrate good quality in the generated images. To show the robustness of our proposed approach, we compared the values obtained from these IoU and DSC measures of the segmented images for each algorithm (LightGBM, SVM, and XGBoost) with our approach based on the stacking generalization method.

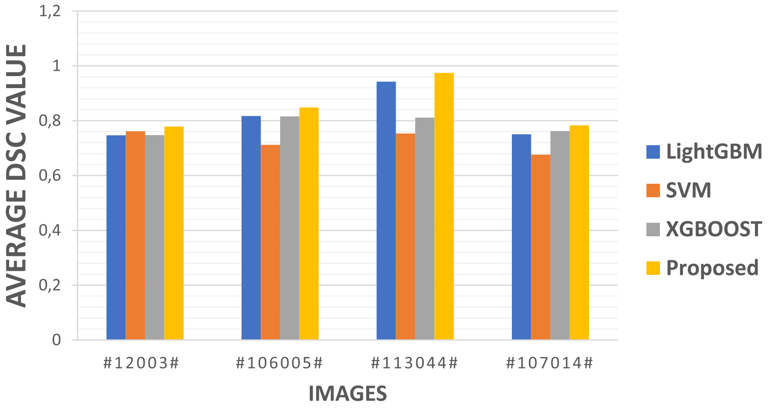


Fig. 8 DSC values obtained by individual's algorithms and our proposed technique

Figures 7 and 8 below present the IoU and DSC values obtained by the individual methods and our stacking ensemble technique respectively on an ensemble of the test images.

From Fig. 7 and according to the results obtained, we can see that our stacking method gave higher values of IoU and DSC metrics compared to those obtained by the other three methods. Precisely instead of choosing a single algorithm, stacking technique allows several algorithms to work together to properly combine the results obtained from the basic model to improve the final prediction.

There are other quantitative assessment techniques such as the following: CC ("correlation coefficient"), SAM ("spectral angular mapper"), SSIM ("structural similarity index measure"), and UQI ("and universal quality index") are used to evaluate the quality of the segmented image, presented in the expressions (5), (6), (7), and (8). Table 2 presents the description and the mathematical expressions of these metrics that we used to evaluate the performance of the segmented images.

Table 3 below represents the values of the metrics for the test images.

The results and the comparison of the different methods obtained from the five measurements CC, SSIM, SAM, and UQI are presented in Table 3 where the best results are highlighted using boldface in each of the rows obtained by the four methods.

Based on the analysis of the results obtained, we compared the proposed approach with the other three models based on a single algorithm (i.e., LightGBM, SVM, and XGBoost) in terms of the values of the five-evaluation metrics presented in the tables above.

For these four test images, we noticed that our stacking proposed framework gives better values of the three metrics CC, SSIM, and UQI depending on the definition of each metric among three machine-learning models. However, these results obtained explain that the predictor model takes advantage of the base model to generate a more robust classifier for better segmentation. We can observe that our proposed framework performs much better than other methods based on a single algorithm in terms of the evaluation metrics. Furthermore, according to Table 3, the SAM metric value is better

Table 2 Performance evaluation metrics used to test the robustness of our proposed approach

Quality metrics	Description	Formula	What value to look for Best fusion (higher/lower)	Reference
CC	Correlation coefficient (CC) indicates the degree of correlation between the enhanced and the original images; it ranges between↰1 and − 1. ↰One and − 1 indicates that the parameters are perfectly correlated, while 0 indicates no relation between two parameters	$CC = \frac{\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \overline{X})^2 \sum_{i=1}^{n}(Y_i - \overline{Y})^2}}$(5) where xi and yi are the pixel values of enhanced image	Higher value (close to + 1)	[51]
SAM	It determines the spectral similarity between the pixels and the vector of the reference and fused images by calculating the spectral angle between these two spectra. It is performed on a pixel-by-pixel base	$SAM(V, \widehat{V})^{\Delta} = arccos\left[\frac{(V, \widehat{V})}{\Vert V \Vert_2 \Vert \widehat{V} \Vert_2}\right]$(6)	Lower value	[52]
SSIM	It calculates the structural similarity index between the reference and fused images, which is a value between − 1 and + 1. A value of + 1 indicates that the 2 given images are very similar or identical	$SSIM = \frac{\left(2\mu_{I_r}\mu_{I_f} + C_1\right)(2\sigma_{I_r}\mu_{I_f} + C_2)}{\left(\mu^2{}_{I_r} + \mu^2{}_{I_f} + C_1\right)(\sigma^2{}_{I_r} + \sigma^2{}_{I_f} + C_2)}$ (7)	Higher value (close to + 1)	[53]
UQI	It is designed to provide a comparison of distortion information between the original image and the distorted image as a combination of three factors, namely loss of correlation, distortion of luminance, and distortion of contrast. UQI is a value between − 1 and + 1. A value of + 1 indicates that the 2 given images are very similar	$UQI = \frac{4 I_r I_f (\mu_{I_r} + \mu_{I_f})}{(\sigma^2 I_r + \sigma^2 I_f)(\mu^2 I_r + \mu^2 I_f)}$(8)	Higher value (close to + 1)	[54]

for first, third, and last image with SVM algorithm, where it gives better result with LightGBM and XGBoost algorithms with the second image. Therefore, with these results obtained, we could say that the proposed approach gave better segmentation results in most of the images tested.

In general, we can say that our framework succeeded in segmenting the test images efficiently and clearly. That can explain the stacking generalization technique can greatly improve the results in terms of the accuracy of image segmentation. Therefore, our proposed framework obtained in all the images of tests the best results of the values of the evaluation metrics except we note that the LightGBM and XGBoost algorithm have the best values of the SAM metric in second test image and SVM with other three images. As a summary, the statistical analysis of experimental results on

Table 3 The performance comparison using different numbers of base learners and the proposed model on test images (best results are highlighted using boldface)

Image	Metrics	LightGBM	SVM	XGBoost	Proposed
12,003	CC	0.0808	− 0.0005	0.0689	**0.1146**
	SAM	0.43924	**0.4383**	0.4412	0.45149
	SSIM	89.2904	89.1634	89.2735	**89.3078**
	UQI	0.2079	0.1939	0.2103	**0.2356**
106,005	CC	− 0.0591	0.0032	− 0.0583	**− 0.0915**
	SAM	**0.4382**	0.4537	**0.4382**	0.4606
	SSIM	**89.3817**	89.3745	89.3823	89.3371
	UQI	0.31457	0.2997	0.3141	**0.3459**
113,044	CC	0.1278	0.0040	0.2279	**0.3979**
	SAM	0.5950	**0.4125**	0.4497	0.6925
	SSIM	89.6365	89.5298	89.6325	**89.3997**
	UQI	0.5901	0.3309	0.3679	**0.6094**
107,014	CC	0.0201	0.0062	0.2704	**0.2760**
	SAM	0.4213	**0.3982**	0.4297	0.4518
	SSIM	89.6396	89.5689	89.6244	**89.6115**
	UQI	0.2994	0.2867	0.3125	**0.3335**

test images shows that our approach obtains better values in terms of metrics for image quality.

Also, to give a subjective evaluation of our proposed approach, we have made a comparison with some recent deep learning-based unsupervised segmentation methods, e.g., NCut [55], CTM [56], JSEG [57], and B-graph [58], and recently proposed algorithms, e.g., Kanezaki [59], W-Net [60], and DC-RM [61] as Fig. 9 shows.

To make a comparison between the methods proposed, we can notice according to the results obtained that Kanezaki approach gives a segmented image whose limits are detected, but it suffers from under segmentation. The W-Net approach segments many regions of low-color contrast, but it does not manage to produce semantically coherent regions. JSEG has its turn segmented many texture regions; on the other hand, it does not work well during regions of low color contrast. CTM works well with larger regions, but this is not the case in texture regions. N-cut over-segments the image and lacks boundary preservation even for similar color regions. On the other hand, the B-graph approach keeps the limits of the object in the segmented image, but it sometimes suffers from under-segmentation problem. DC-RM works well to produce semantically coherent regions; also, it tries to avoid under-segmentation problem. According to Fig. 9, we can also say that our proposed approach segments coherent regions, detected objects, and avoids the under-segmentation problem.

To quantitatively analyze our proposed approach and several other segmentation methods, we utilize segmentation covering (SC) evaluation metric.

Segmentation covering (SC) measures the overlap of regions of the segmentation output and the regions of the ground truth. The higher the SC value, the better the quality of segmentation. It calculates with the following formula:

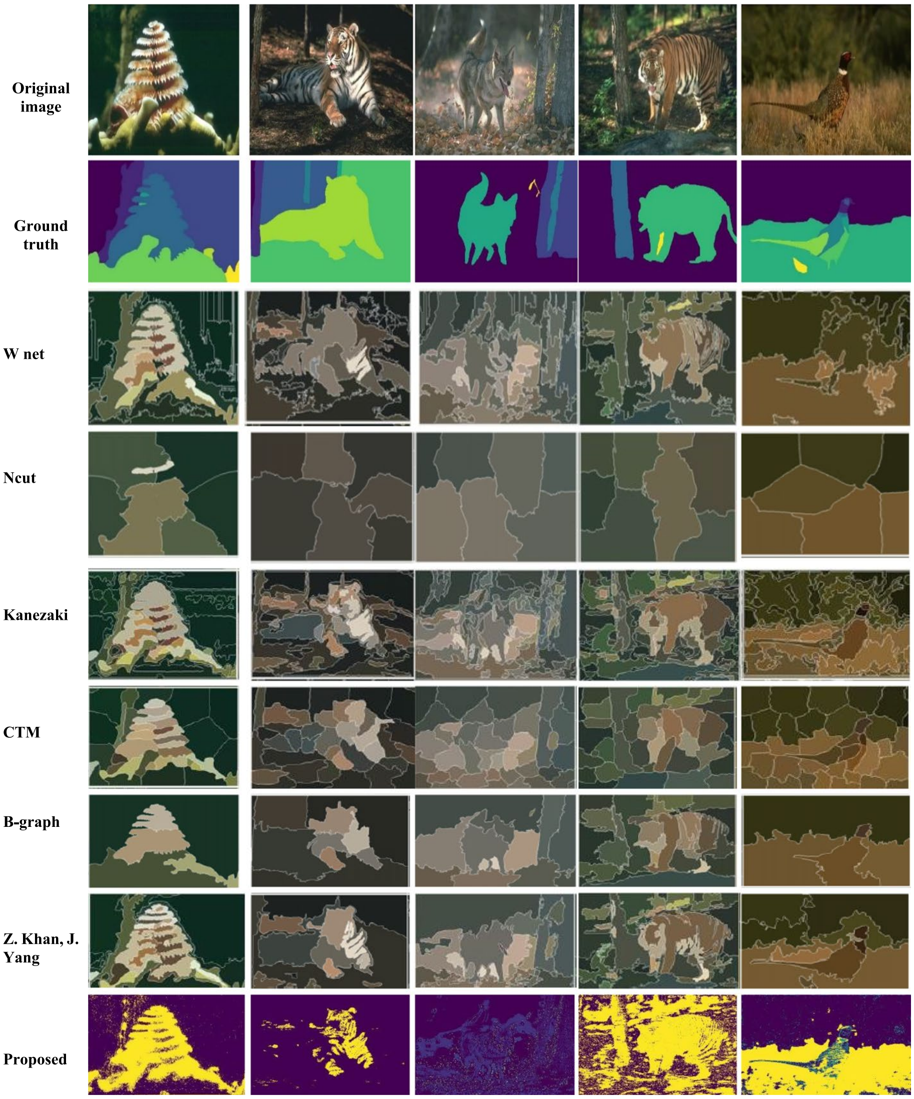


Fig. 9 Segmentation results of various methods for six test images

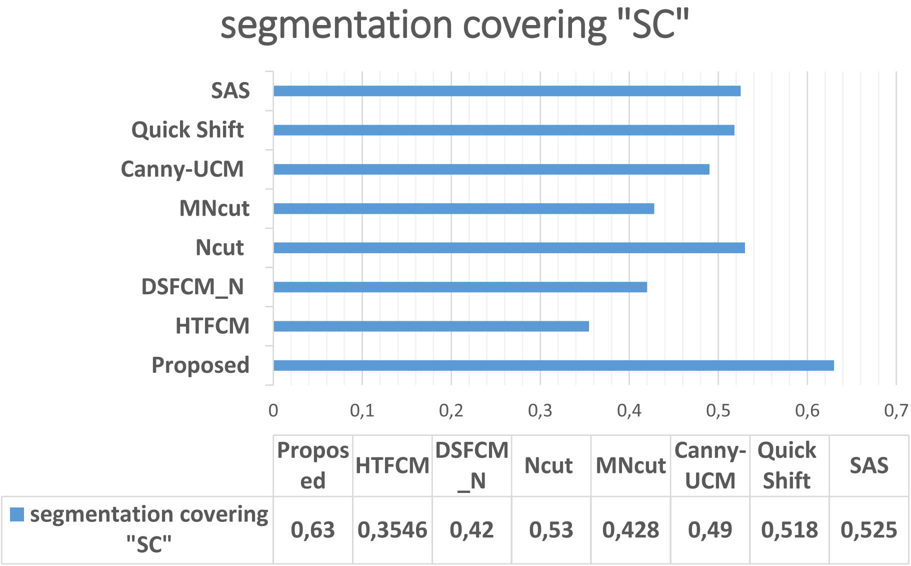
$$c\left(S \to \acute{S}\right) = \frac{1}{N}\sum_{R\in S}|R|.max_{\acute{S}\in S}\varnothing(R,\acute{R})$$

where $\varnothing\left(R,\acute{R}\right) = \frac{\left|R\cap\acute{R}\right|}{\left|R\cup\acute{R}\right|}$ which is basically intersection over union.

Table 4 summarizes the performance of several deep learning-based unsupervised segmentation methods on BSDS500 dataset [62, 63]; from the results obtained, we can say that we have a good covering score compared to the others method mentioned below. Figure 10 presents the results obtained of the score on the covering of the ground truth segments at the BSDS dataset in the form of a diagram.

Table 4 The score on the covering of the ground truth on BSDS500

Methods	**Segmentation covering value** ⇧ **SC**
Proposed	0.63
HTFCM [64]	0.3546
DSFCM_N [65]	0.42
NCut [55]	0.53
MNCut [66]	0.428
Canny-UCM [67]	0.49
Quick shift [68]	0.518
SAS [58]	0.525



	Proposed	HTFCM	DSFCM_N	Ncut	MNcut	Canny-UCM	Quick Shift	SAS
■ segmentation covering "SC"	0,63	0,3546	0,42	0,53	0,428	0,49	0,518	0,525

Fig. 10 Results obtained of the score on the covering of the ground truth segments at the BSDS dataset

Conclusions

In this paper, a stacking ensemble technique is proposed for image segmentation, taking the integration of predictions from two models (SVM and LightGBM) as input to the meta-model (XGBoost in our case) and trying to combine them for the final prediction. The experimental results on the different reference images show that the proposed generalized stack ensemble-learning framework improves the segmentation accuracy compared to the three models based on a single algorithm. Since several research works exist in the literature using different machine learning algorithms, a stacking ensemble is used to obtain a more powerful and robust classifier.

To demonstrate the robustness and the effectiveness of the proposed method, we realized experiments on a set of test images. The result obtained from the individuals' models and our proposed approach are compared using an ensemble of metrics for image quality. The analysis and a comparison of the results obtained showed more consistent predictions for the proposed model. Thus, we have made a comparison with some recent deep learning-based unsupervised segmentation methods. From experimental results, our approach shows that stacked generalization can greatly improve the segmentation effect in terms of accuracy, robustness, and efficiency.

Acknowledgements
Not applicable.

Authors' contributions
FZ as a corresponding author proposed the idea of the paper and wrote the manuscript. FZ, KL, HK, and NEA modeled the system under Python software. Faska et al. contributed to reviewing the paper and have directly participated in the planning, execution, and analysis of this study. All authors read and approved the final manuscript.

Funding
The authors declare that they have no funding for the research.

Availability of data and materials
Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

Declarations

Competing interests
The authors declare that they have no competing interests.

Received: 19 February 2023 Accepted: 17 May 2023
Published online: 07 July 2023

References

1. Khrissi L, El Akkad N, Satori H, Satori K (2022) Clustering method and sine cosine algorithm for image segmentation. Evol Intel 15(1):669–682
2. Khrissi L, Satori H, Satori K, el Akkad N (2021) An Efficient Image Clustering Technique based on Fuzzy C-means and Cuckoo Search Algorithm. Int J Adv Comput Sci Appl 12(6):423–432. https://doi.org/10.14569/IJACSA.2021.0120647.
3. Aliniya Z, Mirroshandel SA (2019) A novel combinatorial merge-split approach for automatic clustering using imperialist competitive algorithm. Expert Syst Appl 117:243–266
4. Javed A, Kim YC, Khoo MC, Ward SLD, Nayak KS (2015) Dynamic 3-D MR visualization and detection of upper airway obstruction during sleep using region-growing segmentation. IEEE Trans Biomed Eng 63(2):431–437
5. Chen X, Williams BM, Vallabhaneni SR, Czanner G, Williams R, Zheng Y (2019) Learning active contour models for medical image segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp 11632–11640
6. Wang X, Wang S, Zhu Y, Meng X (2012) Image segmentation based on support vector machine. In: Proceedings of 2012 2nd International Conference on Computer Science and Network Technology. pp 202–206. https://doi.org/10.1109/ICCSNT.2012.6525921
7. Faska Z, Khrissi L, Haddouch K, El Akkad N (2021) A powerful and efficient method of image segmentation based on random forest algorithm. In: International Conference on Digital Technologies and Applications. Springer, Cham, pp 893–903
8. Khrissi L, El Akkad N, Satori H, Satori K (2020) Image segmentation based on k-means and genetic algorithms. In: Embedded systems and artificial intelligence. Springer, Singapore, pp 489–497
9. Moussaoui H, Benslimane M, El Akkad N (2022) Image segmentation approach based on hybridization between K-means and Mask R-CNN. In: WITS 2020. Springer, Singapore, pp 821–830
10. Gangwar S, Chauhan RP (2015) Survey of clustering techniques enhancing image segmentation process. In: International Conference on Advances in Computing and Communication Engineering. pp 34–39
11. Canny J (1986) A computational approach to edge detection. IEEE Trans Pattern Anal Mach Intell 8(6):679–698
12. Deriche R (1987) Using Canny's criteria to derive a recursively implemented optimal edge detector. Int J Comput Vision 1(2):167–187
13. Karoui I, Fablet R, Boucher J, Augustin J (2010) Variational region-based segmentation using multiple texture statistics. IEEE Trans Image Process 19(12):3146–3156
14. Naz S, Majeed H, Irshad H (2010) Image segmentation using fuzzy clustering: a survey. In: International conference on emerging technologies. pp 181–186
15. Rambabu C, Chakrabarti I, Mahanta A (2004) Flooding-based watershed algorithm and its prototype hardware architecture. IEEE Proc Vision Image Signal Process 151(3):224–234
16. Jiang Y, Zhao K, Xia K, Xue J, Zhou L, Ding Y, Qian P (2019) A novel distributed multitask fuzzy clustering algorithm for automatic MR brain image segmentation. J Med Syst 43(5):118
17. Chen T, Guestrin C (2016) XGBoost: a scalable tree boosting system. In: Proceedings of the 22Nd ACM SIGKDD international conference on knowledge discovery and data mining. pp 785–794. https://doi.org/10.1145/2939672.2939785
18. Yang XD, Wang JM, Zhang LN (2017) Application of XGBoost in ultra-short term load forecasting. Electr Drive Autom 39:21–25
19. Zhao T, Zheng S, Li W (2018) Research on credit risk analysis based on XGBoost. Softw Eng 21:33–35

20. Wang C, Deng C, Wang S (2019) Imbalance-XGBoost: leveraging weighted and focal losses for binary label-imbalanced classification with XGBoost, arXiv. Available online: https://arxiv.org/abs/1908.01672. Accessed 24 Jan 2021
21. Zhu S, Zhu F (2019) Cycling comfort evaluation with instrumented probe bicycle. Transp Res Part A Policy Pract 129:217–231. https://doi.org/10.1016/j.tra.2019.08.009
22. Mo H, Sun H, Liu J, Wei S (2019) Developing window behavior models for residential buildings using XGBoost algorithm. Energy Build 205:109564. https://doi.org/10.1016/j.enbuild.2019.109564
23. Yue L, Yi Z, Pan J, Li X, Li J (2021) Identify M subdwarfs from M-type Spectra using XGBoost. Optik 225:165535. https://doi.org/10.1016/j.ijleo.2020.165535
24. Reinstein I (2017) XGBoost a top machine learning method on Kaggle, explained. Available online: http://www.kdnuggets.com/2017/10/xgboost-topmachine-learning-method-kaggle-explained.html. Accessed 23 Jan 2021
25. Tianqi C, Guestrin C (2016) XGBoost: a scalable tree boosting system. In: Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining. pp 785–794
26. Wang L, Wu C, Tang L, Zhang W, Lacasse S, Liu H, Gao L (2020) Efficient reliability analysis of earth dam slope stability using Extreme Gradient Boosting method. Acta Geotech 15(11):3135e3150
27. Ke GL, Meng Q, Finley T, Wang TF, Chen W, Ma WD, Ye QW, Liu TY (2017) LightGBM: a highly efficient gradient boosting decision tree. In: Proceedings of the 31st Annual Conference on Neural Information Processing Systems, Long Beach, CA, USA. pp 3146–3154
28. Vapnik V, Cortes C (1995) Support-vector networks. Mach Learn 20:273e297
29. Vapnik VN (1998) Statistical learning theory. Wiley, New York
30. Zhang C, Chen X, Chen M, Chen S-C (2005) A multiple instance learning approach for content based image retrieval using one-class support vector machine. In: Proceedings of the IEEE International Conference on Multimedia and Expo. pp 1142–1145
31. Zhang L, Lin F, Zhang B (2001) Support vector machine learning for image retrieval. In: Proceedings of the IEEE International Conference on Image Processing. pp 721–724
32. Shao H, Jiang H, Lin Y, Li X (2018) A novel method for intelligent fault diagnosis of rolling bearings using ensemble deep auto-encoders. Mech Syst Signal Process 102:278–297
33. Re M, Valentini G (2012) Ensemble methods: a review. In: Advances in Machine Learning and Data Mining for Astronomy. London, United Kingdom: Chapman & Hall.
34. Dietterich TG (2000) Ensemble Methods in Machine Learning. In: Multiple Classifier Systems. MCS 2000. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, Vol. 1857 pp 1–15. https://doi.org/10.1007/3-540-45014-9_1.
35. Zhou J, Peng T, Zhang C, Sun N (2018) Data pre-analysis and ensemble of various artificial neural networks for monthly streamflow forecasting. Water 10:628
36. David B (2018) Online cross-validation-based ensemble learning. Stat Med 2:37
37. Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, Ye Q, Liu TY (2017) LightGBM: a highly efficient gradient boosting decision tree. In: 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA
38. Hu CA, Chen CM, Fang YC, Liang SJ, Wang HC, Fang WF, Sheu CC, Perng WC, Yang KY, Kao KC, Wu CL, Tsai CS, Lin MY, Chao WC (2020) Using a machine learning approach to predict mortality in critically ill influenza patients: a cross-sectional retrospective multicentre study in Taiwan. BMJ Open 10(2):e033898
39. Website of the Kaggle. Available online: https://www.kaggle.com/. Accessed 15 Dec 2021
40. Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, Ye Q, Liu T-Y (2017) LightGBM: a highly efficient gradient boosting decision tree. In: Advances in neural information processing systems. pp 3146–3154. http://www.audentia-gestion.fr/MICROSOFT/lightgbm.pdf
41. Kodaz H, Özşen S, Arslan A, Güneş S (2009) Medical application of information gain based artificial immune recognition system (AIRS): diagnosis of thyroid disease. Expert Syst Appl 36:3086–3092
42. Mountrakis G, Im J, Ogole C (2011) Support vector machines in remote sensing: a review. ISPRS J Photogramm Remote Sens 66:247–259
43. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20:273–297
44. Wolpert DH (1992) Stacked generalization. Neural Netw 5:241e259. https://doi.org/10.1016/S0893-6080(05)80023-1
45. Ren Y, Zhang L, Suganthan PN (2016) Ensemble classification and regression-recent developments, applications and futuredirections. IEEE Comput Intell Mag 11(1):41–53
46. Mitchell TM, Keller RM, Kedar-Cabelli ST (1986) Explanationbased generalization: a unifying view. Mach Learn 1(1):47–80
47. Canny J (1989) A computational approach to edge detection. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-8(6). pp 679–698
48. Arbelaez P, Maire M, Fowlkes C, Malik J (2011) Contour detection and hierarchical image segmentation. IEEE Trans Pattern Anal Mach Intell 33:898–916
49. Kumar P (2019) Machine learning quick reference. Packt Publishing Ltd., Birmingham
50. Hsu C, Chang C, Lin C (2003) A practical guide to support vector classification. pp 1–16
51. Yilmaz V, Gungor O (2016) Determining the optimum image fusion method for better interpretation of the surface of the Earth. Nor Geogr Tidsskr 70(2):69–81
52. Alparone L, Wald L, Chanussot J, Member S, Thomas C, Gamba P, Bruce LM (2007) Comparison of pansharpening algorithms: outcome of the 2006 GRS-S data-fusion contest. IEEE Trans Geosci Remote Sens 45:3012–3021
53. Wang Z, Bovik AC, Sheikh HR, Member S, Simoncelli EP, Member S (2004) Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 13:1–14
54. Alparone L, Aiazzi B, Baronti S, Garzelli A, Nencini F, Selva M (2008) Multispectral and panchromatic data fusion assessment without reference. Photogramm Eng Remote Sens 74:193–200. https://doi.org/10.14358/PERS.74.2.193
55. Shi J, Malik J (2000) Normalized cuts and image segmentation. IEEE Trans Pattern Anal Mach Intell 22(8):888–905
56. Yang AY, Wright J, Ma Y, Sastry SS (2008) Unsupervised segmentation of natural images via lossy data compression. Comput Vis Image Underst 110(2):212–225

57. Deng Y, Manjunath B (2001) Unsupervised segmentation of color-texture regions in images and video. IEEE Trans Pattern Anal Mach Intell 23(8):800–810
58. Li Z, Wu X-M, Chang S-F Chang (2012) Segmentation using superpixels: A bipartite graph partitioning approach, in Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE 789–796.
59. Kanezaki A (2018) Unsupervised Image Segmentation by Backpropagation. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada 1543-1547. https://doi.org/10.1109/ICASSP.2018.8462533.
60. Xia X, Kulis B (2017) W-Net: a deep model for fully unsupervised image segmentation, arXiv preprint arXiv:1711.08506
61. Khan Z, Yang J (2020) Bottom-up unsupervised image segmentation using FC-Dense U-Net based deep representation clustering and multidimensional feature fusion-based region merging. Image Vis Comput 94:1–11. https://doi.org/10.1016/j.imavis.2020.103871Elsevier
62. Zhang Y, Zhang H, Guo Y, Lin K, He J (2019) An adaptive affinity graph with subspace pursuit for natural image segmentation. In: 2019 IEEE International Conference on Multimedia and Expo (ICME), Shanghai, China. pp 802–807
63. Donoser M, Schmalstieg D (2014) Discrete-continuous gradient orientation estimation for faster image segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 3158–3165
64. Tan KS, Ashidi MIN (2001) Color image segmentation using histogram thresholding—fuzzy C-means hybrid approach. Pattern Recognit 44:1–15
65. Zhang YX, Bai XZ, Fan RR, Wang ZH (2019) Deviation-sparse fuzzy C-means with neighbor information constraint. IEEE Trans Fuzzy Syst 27(1):185–199
66. Cour T et al (2005) Spectral segmentation with multiscale graph decomposition. IEEE Conf Comput Vision Pattern Recognit 2:1124–1131
67. Arbelaez P (2006) Boundary extraction in natural images using ultra-metric contour maps. In: IEEE Conference on Computer Vision and Pattern Recognition Workshop. pp 182–182
68. Vedaldi A, Soatto S (2008) Quick shift and kernel methods for mode seeking. In: European Conference on Computer Vision. pp 705–718

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.