Advances in Artificial Intelligence and Machine Learning; Research 4 (4) 3135-3160 Received 16-09-2024; Accepted 22-12-2024; Published 29-12-2024

A Random Forest and Edge Vector Ensemble Model for Segmenting Aerial Satellite Forest Images

Trust Tshepo Mapoka
Department of Computer Science, University of Botswana
Gaborone, Botswana

Clopas Kwenda
Department of Informatics, Namibia University of Science and Technology
Namibia

Wilson Bakasa bakasaw@ub.ac.bw
Department of Computer Science, University of Botswana
Gaborone, Botswana

Corresponding Author: Wilson Bakasa

Abstract

The attribute of image segmentation significantly impacts the validity of the resulting classification, making it an essential step in the image classification process. Present segmentation methods cannot produce a feature set that yields a segmented image of good quality. This work creates a method that yields a set of believable attributes and produces segmented images of excellent quality. The authors aim to create a novel machine-learning model to enhance the image segmentation quality of aerial satellite images using metrics such as Intersection Over Union (IoU), Receiver Operating characteristic (ROC) curves, and accuracy. The Random Forest (RF) algorithm-based machine learning model is intended to separate forested and non-forest regions from aerial satellite images. Finding edges and separating layered objects are two computer vision problems that can be solved using RF, a supervised machine learning model. Our method objectively assesses the quality of image segmentation by examining the places at which all image objects overlap with the real image regions of a scene item. After generating a collection of features, the RF performed the segmentation process using the Gabor filters and edge detection techniques, such as the Canny and Sobel filters. Segmented images were compared against real masks. The suggested model's superior segmentation capability, with 90% accuracy, is evaluated against several baseline algorithms, including Linear Discriminant Analysis (LDA), Linear Support Vector Machine (LSVM), and Gaussian Naive Bayes (GNB). For SVM, GNB, and LDA, the corresponding accuracy rates are 81%, 89%, and 85%.

Keywords: Segmentation, Random forest, Supervised approach, Edge vector, Gabor filter.

Citation: Trust Tshepo Mapoka, et al. A Random Forest and Edge Vector Ensemble Model for Segmenting Aerial Satellite Forest Images. Advances in Artificial Intelligence and Machine Learning. 2024;4(4):180.

1. INTRODUCTION

When it comes to image processing, image segmentation is necessary as it subsequently leads to image classification, however conducting such as process is a very difficulty task [1]. An image is divided into groups of features with comparable pixels by a technique known as image segmentation; the line that divides the group is called an edge [2]. An ideal segmentation algorithm perfectly matches the partitioned or segmented images with the actual image regions [3]. However, there is some mismatch, which determines the segmentation quality by the respective segmentation algorithm. The quality of segmentation hugely affects subsequent image classification results. The primary function of all segmentation algorithms is the ability to capture essential features referenced by segments such that they reflect global aspects of the image [4]. Spatial context relates pixels to objects [5]. Given cognisant of human experts' effectiveness in performing object detection and image segmentation compared to computer vision techniques, performing such tasks would take a long time and becomes unsustainable when handling large volumes of data [6]. Therefore, such cases drive the need to rope in computer vision algorithms, though they pose challenges in determining real-segmentation findings. The opaque nature of image visual patterns in image perception makes it difficult [7]. Algorithms for evaluating image segmentation fall into two main groups: supervised and unsupervised. A segmented image is compared to a reference image known as the ground truth using supervised evaluation techniques, also known as empirical discrepancy approaches [8]. The degree to which the segmented image resembles the ground truth measures the segmentation method's quality [9]. Because they offer a finer evaluation resolution, supervised evaluation techniques are dependable because of this direct comparison. However, since different specialists provide different reference images, creating ground truth is still challenging and subjective [10]. According to the specified features that characterise a well-segmented image, unsupervised algorithms assess a segmented image [8, 11, 12]. Consequently, these characteristics are the foundation for evaluating how well-unsupervised algorithms work. Since unsupervised techniques don't need reference images, they are also known as stand-alone or empirical goodness techniques. Four criteria were provided by Haralick and Shapiro [13] to determine what makes segmentation good.

- Characteristics of a region are to be uniform and homogeneous
- There should be sharp differences between regions in terms of that characteristic, which is uniform
- Interior of a region should be simple and without holes
- Boundaries should be clear and not ragged

Supervised methods are far more preferred for segmentation evaluation in every situation where a trustworthy ground truth image has been produced, claims Cheng [14]. As a result, this work proposes a model that uses the supervised technique to segment aerial satellite images. These images were extracted from the reference image-containing Kaggle data set.

Furthermore, there exist top-down, bottom-up, and hybrid approaches to image segmentation. The top-down segmentation technique uses prior knowledge of an object to predict plausible pixels that will be used to create an object. However, the approach suffers many variabilities regarding object

appearance and shape for a particular category. In the same vein, extensive experiments have demonstrated that an object generated by a top-down segmentation technique cannot even match the corresponding physical object [15]. To overcome the drawbacks of this strategy, a bottom-up method was suggested. This method attempts to find the labelled objects that correspond to the many sub-regions that are created by first partitioning an image. The bottom-up approach depends on continuity principles and low-level features such as colour, SIFT, and texture. However, this approach cannot capture high-level features such as shape. Hence, an object might be segmented into multiple unnecessary regions. A hybrid approach of the bottom-up and top-down approaches through random fields has been proposed, but the model only focused on encoding local image properties. Canny, Sobel, and Gabor filters are widely used in image segmentation. Gabor filters are beneficial for detecting features such as textures, patterns, and edges because they analyse the frequency and orientation of edges in an image [16]. Sobel filters are utilised to detect image edges by computing the intensity gradient, and therefore, they can be used for edge detection, image enhancement, and noise reduction [17]. For edge detection, Canny filters compute the gradient of the image's intensity, suppress noise, and identify edge pixels through thresholding and hysteresis [17]. Gabor filters can segment an image into regions with comparable textures or patterns. Sobel filters can isolate and enhance the margins and boundaries that define image regions. Canny filters can extract the edges and boundaries of image regions, making them useful for segmenting objects with clearly defined edges. A hybrid of these features aids in increasing the quality of image segmentation. The choice of using traditional algorithms like the Gabor filter, Sobel filter, and Canny filter is often driven by their specific strengths in texture analysis, edge detection, computational efficiency, and robustness to noise. These reasons make them valuable tools in the image processing toolkit, even as newer methods continue to be developed [18–20]. It is for this reason this study proposes to build an ensemble of Canny, Sobel, and Gabor filters that can effectively extract key image features for subsequent image segmentation.

In this study, we overcome the issues mentioned above by putting forward an RF-based ensemble edge vector feature identification model. To assess the model's performance, we used ROC graphs, the Jaccard index, and accuracy. According to the experimental results, the proposed model performed better than the others in segmentation, with an accuracy of 90%, as opposed to 81%, 89%, and 85% for LSVM, GNB, and LDA, among others.

The study's contribution is summarized as follows:

- Utilizing a combination of Gabor, Canny, and Sobels filters, we present a novel approach to image segmentation. This method integrates features taken from the three filters to create a complete collection of features that includes all information required for segmentation. For ensuing segmentation tasks, machine learning classifiers use these integrated features.
- Due to their distinct advantages in texture analysis, edge detection, computing efficiency, noise resistance, and robustness to conventional machine learning classifiers, Gabor, Canny, and Sobels are combined in this innovative hybrid model to provide better aerial satellite image segmentation. Each classifier employs different approaches and learning paradigms to bring specific capabilities to the segmentation problem.

This is how the document is structured: Section 2 reviews relevant studies. A summary of feature extraction algorithms is given in Section 3, while the segmentation process with machine learning

classifiers is looked at in Section 3. The proposed model's framework is shown in Section 5. Section 6 discusses the findings, and the study's conclusion is given in Section 7. Future work is covered in section 8.

2. RELATED STUDIES

To determine whether or not the pixels in the image are in a forest, the model uses a collection of decision trees using the Random Forest approach. An edge vector is also included in the model to increase segmentation accuracy. Prior research [21–25] that employed methods similar to those in the suggested model to segment remote sensing images served as the foundation for the suggested model. Random Forest is one of the machine learning methods for image segmentation that has lately gained prominence because of its capacity to handle complicated attributes and enormous datasets.

For categorising high-resolution satellite photos of the Amazon jungle, a scalable and accurate method is proposed in [26]. The weighted voting of heterogeneous classifiers is used to implement the ensemble-based approach. The ensemble learner simultaneously trains Five distinct classification models using a publicly available dataset of user-labeled images. Two meta-classifiers are created by adding further features to the original model's predictions. Individual classifier performance is compared with ensemble performance using a holdout test dataset.

Semantic land cover segmentation is used by the machine learning method Random Forest to determine farmland boundaries and estimate agricultural acreage [27]. Next, features are determined by applying various filters to a machine-learning model that incorporates the chosen attributes. The satellite images in the collection were taken using the QGIS program. Agricultural mapping may benefit from Random Forest's superior performance over previous techniques, as seen by its 90% test accuracy and 99% training accuracy.

One method that separates remote sensing image data into forests is called SegForest. The weight-based cross-entropy loss function (WBCE), multi-feature fusion (MFF), and multi-scale multi-decoder (MSMD) are its three extra modules [28]. It achieved a 91% mean accuracy and an 83.39% mean intersection over union on DeepGlobe-Forest and Loveda-Forest, respectively, compared to other sophisticated segmentation techniques. The experiment outcomes demonstrate SegForest's effectiveness and the generalization of issues about the segmentation of forest remote sensing images.

Using data from unmanned aerial vehicles from 2013 to 2023, Chehreh et al.'s study looks at the classification and segmentation of trees. It emphasises segmentation and classification tasks employing RGB, multispectral, and LiDAR scanners [29]. This study focuses on machine learning techniques, data and remote sensing methodologies, and recent research advances. Unmanned aerial vehicle-based automatic tree classification and segmentation advances are summarised in detail in this thorough review of 979 publications, comprising 144 articles. The importance of image processing and machine learning in enhancing interpretability and accuracy in these tasks is emphasised in the study.

The paper's ensemble of deep learning (DL) and machine learning (ML) architecture classifies the vegetation in Eastern Serbia using a biased Support Vector Machine (B-SVM), Random Forest (RF), and Convolutional Neural Network (CNN) according to Drobnjak et al. The ensemble architecture outperformed CNN, RF, and B-SVM, according to modelling results, with a total accuracy of 0.93 [30]. Next were RF, CNN, and B-SVM achieving 0.91, 0.90, and 0.88 respectively.

Guo et al. developed an effective system for identifying urban trees using a random forest method and object-oriented design. From UAV multispectral images, the segments were extracted using a multi-scale segmentation technique with visual discrimination. Using geometric, textural, index, and spectral components, schemes S1–8 and S9 were constructed. RF, K-nearest neighbour (KNN), and Support vector machines (SVM) classifiers were used to classify urban trees [31]. The accuracy of the RF classifier was the highest, with S9 outperforming SVM and KNN. Spectral features were more important than geometric features, which harmed classification. However, Camphor and Cinnamomum Japonicum's ambiguous data suggest that more traits, including height, should be added in the future.

Table 1: Summary of related work.

Study or Model	Technique	Dataset/ Images	Accuracy or Performance	Additional Modules/ Features	Reference
Random Forest Model	Random Forest	Satellite images of Amazon	Scalable, accurate method	Ensemble-based, weighted voting of heterogeneous classifiers	[26]
Cropland Boundaries	Random Forest	Satellite images (QGIS)	99% training, 90% test accuracy	Various filters for feature identification	[27]
SegForest	Random Forest with WBCE, MSMD, and MFF	DeepGlobe-Forest, Loveda-Forest	mIoU: 83.39%, mAcc: 91.00%	multi-scale multi-decoder (MSMD), Weight based cross entropy (WBCE) loss function, multi feature fusion (MFF)	[28]
Chehreh et al.	Machine learning with RGB, multispectral, LiDAR	UAV images (2013-2023)	-	Highlighting trends in segmentation and classification	[29]
Drobnjak et al.	Ensemble DL and ML (CNN, RF, B-SVM)	Vegetation in Eastern Serbia	Accuracy: 0.93	RF, CNN, B-SVM	[30]
Guo et al.	Random Forest with multi-scale segmentation	UAV multispectral images	Best accuracy with S9 scheme	Spectral, index, textural, geometric features; compared with KNN and SVM classifiers	[31]

3. OVERVIEW OF FEATURE EXTRACTION ALGORITHMS

The study's techniques for extracting the features needed for further segmentation are summarised in this section. The Gabor, Canny, and Sobel filters are specifically covered in this section.

3.1 Gabor Filter

Dennis Gabor first introduced Gabor filters. Their capacity to gather information about orientation selectivity, spatial locality, and frequency characteristics has led to their widespread use in computer

vision and image analysis tasks [9]. Gabour filter is widely used for texture analysis and works by analysing specific frequency content in an image along a particular direction within the analysis region [32]. This filter is appropriate for this study as forest regions have different textures from non-forest areas. A sinusoidal plane wave with a certain frequency and orientation (carrier) that comprises a Gabor function f(m,n) is modulated by a two-dimensional translated Gaussian envelope [33].

$$f(m,n) = s(m,n) * w(m,n) \tag{1}$$

Orientation is expressed as follows:

$$s(m,n) = cos(2\pi.(y_0.m + z_0.n) + \phi) + i.sin(2\pi.(y_0.m + z_0.n) + \phi) \tag{2}$$

where (y_0, z_0) denotes spatial frequency of complex sinusoid and ϕ denotes the phase of the complex sinusoid. The Gaussian envelope function is expressed as:

$$w(m,n) = k.exp\left\{-\pi\left(\frac{(m-m_0)_r^2}{\sigma_m^2} + \frac{(n-n_0)_r^2}{\sigma_n^2}\right)\right\} \tag{3}$$

Where k is the Gaussian envelope magnitude, σ_m and σ_n are the Gaussian envelope scaling parameters, (m_0, n_0) represents the peak of the Gaussian function, the subscript r in $(n-n_0)_r$ and $(m-m_0)_r$ denotes the coordinates after rotation, θ is the rotation angle of the Gaussian envelope where:

$$(m-m_0)_r = (m-m_0)cos\theta + (n-n_0)sin\theta \tag{4}$$

$$(n-n_0)_r = (m-m_0)sin\theta + (n-n_0)cos\theta \tag{5}$$

Given that y_0, z_0 represent the spatial frequency of complex sinusoid, the orientation is defined θ as

$$\theta = tan^{-}\frac{y_0}{z_0} \tag{6}$$

and the radial center f_0 as

$$f_0 = \sqrt{(y_0^2 + z_0^2)}. \tag{7}$$

The Gabor filter function f(l,k) applied on an image r(l,k) transforms image r(l,k) into image i(l,k) expressed as:

$$i(l,k) = f(l,k) * r(l,k) = \sum_{b=0}^{B-1}\sum_{c=0}^{C-1} f(b,c).r(l-b,k-c) \tag{8}$$

Where r(m,n) is the pixel value of the input images at location (m,n), * denotes a 2-dimensional linear convolution, and B and C are the sizes of the Gabor filter mask. By computing the absolute values between the filtered images and the μ value of the window W of size $B_m B_n$, the energy of the filtered image E(l,k) is determined. The expression for E(m,n) is:

$$E(m,n) = \frac{1}{B}\sum_{(a,b)\in W} |t(a,b) - \mu| \tag{9}$$

3.2 Canny Edge Detector

Most edge detection methods, such as those developed by Sobel, Prewitt, and Roberts [34], are based on the gradient concept. Finding the proper edge, turnover point, and slope location is the

primary difficulty with these methods. The Laplacian methodology, which determines the correct edge using the second derivative, is more practical. The Laplacian approach is the foundation of the Canny detection algorithm. The method focuses on the gradient's size, direction, and intensity in the image. The definition of gradient image intensity is:

$$\Delta f = \left[\frac{\delta f}{\delta m} + \frac{\delta f}{\delta n}\right]^t = [V_m, V_n] \tag{10}$$

Magnitude M is expressed as:

$$M = \sqrt{V_m^2 + V_n^2} \tag{11}$$

The direction of the gradients is expressed as:

$$\theta = tan^{-1}\frac{V_m}{V_n} \tag{12}$$

The Canny technique relies on determining the squared gradient magnitude to locate edges, as shown in equation 23. Edges are defined as gradient values above a certain threshold.

3.3 Sobel Filter

An additional method for detecting edges is the Sobel filter. In a given image, the method computes the image intensity gradient of each pixel value. The biggest gradient rise from light to dark determines the direction. Additionally, the orientation of the edge is specified by the Sobel filter. Two 3 x 3 kernels are used in its operational function [35]. Changes in the horizontal direction are accommodated by one, while those in the vertical direction by the other. The derivative approximations are computed by convolving these two kernels on the original image. Assume that two images with horizontal and vertical approximations are V_x and V_y. The computation is expressed as follows:

$$V_x = \begin{bmatrix} 2 & 1 & -2 \\ 3 & 1 & -3 \\ 2 & 1 & -2 \end{bmatrix} \times I$$

$$V_y = \begin{bmatrix} -2 & -3 & -2 \\ 1 & 1 & 1 \\ 2 & 3 & 2 \end{bmatrix} \times I$$

Where I is the image source.

4. OVERVIEW OF MACHINE LEARNING TECHNIQUES FOR SEGMENTATION

The machine learning classifiers that were employed to complete the segmentation challenge are described in this section. The RF algorithm, GNB algorithm, LDA technique and LSVM method are among the machine learning classifiers covered in this section.

4.1 Random Forest

An ensemble classifier, the random forest algorithm gets its name from how its decision trees grow at random [36]. Decision trees are an efficient way to process large amounts of data. The method randomly selects a sample of training data throughout the training phase. The split function and threshold are used to iteratively divide the training data into left and right subsets at a given node, let's say n. The vector v split function, $X(v_i)$, selects the threshold at random from the range $h \in (minX(v_i), maxX(v_i)$, where h is the threshold. The split function that generates the left and right subset trees has the following expression:

$$q_l = (z \in q_s)|x(v_z) < h) \tag{13}$$

$$q_r xs = q_s \backslash q_z \tag{14}$$

where q_l represents the left data, q_r represents the right data, and q_s represents the data at the corresponding node n. Several candidates are randomly generated at the split node using the threshold and the split function. Candidates who maximise the information obtained at a specific node are chosen. The foundation for calculating the information gain is provided by equation 15.

$$\Delta I = -\frac{|q_z|}{|q_s|}E(q_s) - \frac{|q_r|}{|q_s|}I(q_s) \tag{15}$$

where the knowledge gain is evaluated as ΔI. The iterative procedure ends when ΔI becomes a dead node. Equation 16 concludes the last class with the ensemble of all dispersed trees, Y = $(y_1, y_2, ..., y_s)$.

$$P(c_z|Y) = \frac{1}{W}\sum_{s=1}^{W} P(c_z|x_W) \tag{16}$$

where, given distributed trees Y, $P(c_z|Y)$ is the probability of class c_z.

4.2 Gaussian Naive Bayes

Based on the premise that characteristics of specified classes are independent of one another, Gaussian Naive Bayes (GNB) was developed [37]. The GNB is allocated to a class C in a specific testing instance m using the formula in 17.

$$P(C) = \prod_{i}^{x} = I_p(M_i|C) \tag{17}$$

where C represents the class where the instance m belongs, and M = $(M_1, ..., M_x$ denotes a feature vector.

4.3 Linear Discriminant Analysis

Using the formula in 18, the Linear Discriminant Analysis (LDA) distinguishes between input patterns using a decision boundary that divides two groups. [37].

$$d(I) = I_1 - xI_2 - r \tag{18}$$

Input patterns are indicated by I_1 and I_2. The patterns of one class are covered by $d(I) > 0$, whereas the patterns of another class are covered by $d(I) < 0$. Think about the situation where b classes are provided, and the task is to assign a test example f to any of the classes. Class G, not H, would be given the test example f if

$$d_g(f) > d_h(f) \tag{19}$$

where $g \neq h$ if g,h = 1,2,...,s. The decision boundary that distinguishes classes g and h is expressed as

$$d_g(f) - d_h(f) = 0 \tag{20}$$

hence, the test example f would be assigned to class g if

$$d_g(f) - d_h(f) > 0 \tag{21}$$

and into class h if

$$d_g(f) - d_h(f) < 0 \tag{22}$$

4.4 Linear Support Vector Machine

Given a dataset $\{(m_1, n_1)...(m_n, n_n)\}$ the objective is to construct a large-margin classifier that effectively separates the two data classes. This classifier is defined by a hyperplane, which distinguishes the points according to the equation provided in 23

$$\overrightarrow{w}.\overrightarrow{m} - b = 0 \tag{23}$$

The normal of the hyperplane is represented by the vector $\overrightarrow{w}$. The soft and hard margins are the two types of margins that define the hyperplane. When training data is linearly separable and error-free (i.e., free of noise or outliers), the hard margin method is employed. However, when errors are present, they can result from either a reduced margin or the inability of the hard margin to separate the data properly. The process of constructing the hard margins follows these steps:

- All points located outside the margins are represented as $W^T m_j + b \geq a$
- The Maximisation of the margin is expressed as

$$Max\gamma = a\ ||w|| \tag{24}$$

- Lastly, when a is set to 1, the minimum obtained is expressed as $||w||(w^{(t)}.m_j) + b \geq 1$

The dual form is generated when w is represented as a linear combination of the training observations. This occurs when $w = \sum \alpha_i n_i m_i$ where the coefficient α is non-zero only for the support vectors. For nonseparable (overlapping) classes, Linear SVM introduces slack variables ζi and ζj to account for classification errors. The optimisation problem becomes minimising $min||w||$ while incorporating a penalty for misclassification, represented by $c \sum \zeta j$. This is subject to the constraint $(W^T.m_j + b)n_j \geq I - \zeta j$, which ensures that the slack variable ζj quantifies the degree of misclassification. Linear SVM works by identifying two parallel hyperplanes that separate two classes to maximise the distance between them. The scaling of the datasets is represented by equations 25 and 26, which label the classes based on their separation by the boundary. Any point

that falls above the boundary defined by equation 25 belongs to one class, while points below the boundary defined by equation 26 belong to the other class.

$$\overrightarrow{w}.\overrightarrow{m} - b = 1 \tag{25}$$

$$\overrightarrow{w}.\overrightarrow{m} - b = -1 \tag{26}$$

5. MATERIALS AND METHODS

5.1 Data Set

According to [38], the experiment's dataset was acquired from Kaggle. Partitioned into training and test sets of 643/249 images each (or 80%/20% segmentation), the dataset consists of 803 RGB satellite shots. Digital Globe's satellite captured the imagery with a pixel resolution of 50 cm and an image space resolution of 2448 × 2448. Each satellite image is linked with a mask image to annotate forest cover. An RGB mask, or a coloured image with the same height and width as the input image, is what the solution is supposed to predict for the input. The trial was conducted using the GPU and TPU cloud resources freely available in the Google Colab environment. Because of the experiment's high computing requirements, the GPU was specifically used in conjunction with NVIDIA Tesla acceleration. TABLE 2, contains the specifications for the hardware and software, whereas TABLE 3, displays the computational time for each machine classifier for training and prediction times.

Table 2: Software and Hardware requirements used in the model

Software	**Hardware**
Python version 3.9	CPU: core i5 2.2 GHZ
Backend: The TensorFlow GPU	RAM: 32 GB
Application Programming Interface: The Keras GPU	GPU
	HDD: 500 GB
	NVIDIA, 16 GBs RAM

Table 3: Computational time for each classifier

Classifier	**Training Time (seconds)**	**Prediction Time (seconds)**
RF	252	17
LinearSVM	260	21
GNB	480	29
LDA	320	26

5.2 The Proposed Model

In FIGURE 1, the suggested model for segmenting forest images is displayed. For the RF to complete the segmentation task, the model generates believable features using the Gabor filter,

Prewitt, and Canny edge detection techniques. The fact that Gabor filters are highly successful in

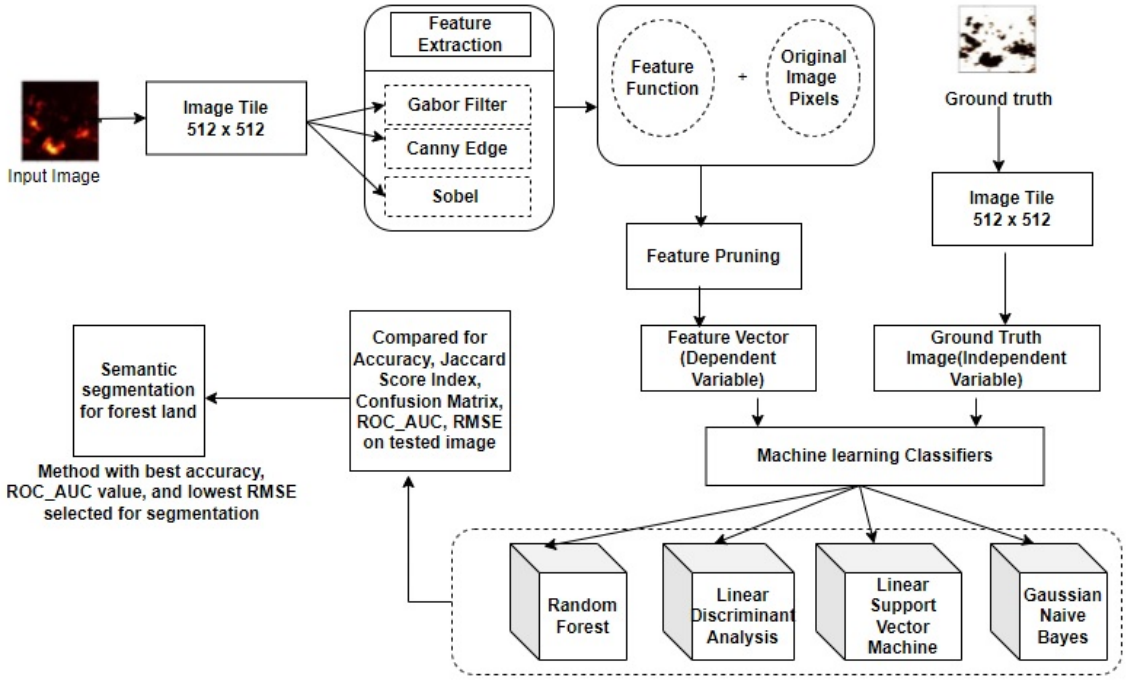

Figure 1: Proposed segmentation model

extracting low-level features for all image processing issues supports the choice of the Gabor filter [39]. The effectiveness of edge detection methods was tested in a study [40]. The results concluded that the Canny algorithm, which used the Gaussian technique, generated the best segmentation results, and Sobel was the best of the methods that performed the edge detection. Thus, the study employed an ensemble stacking technique of Gabor filter and edge detection algorithms to create a set of features needed for further segmentation. This study used the Random Forest (RF) method for the segmentation process. The RF technique functions effectively in object detection for multi-spectral images, according to another study by [41]. Other algorithms, including Support Vector Machines (SVM), Gentle AdaBoost (GAB), and Maximum Likelihood Classification (MLC), were beaten by the RF [42]. This study opted to use the RF method in light of these circumstances.

5.2.1 Feature function

Feature outputs from Gabor, Sobel, and Canny filters are used as inputs to a random classifier for image segmentation tasks. The process proceeds as follows:

- Step 1: The Gabor filter is utilized at each pixel to compute the filtered image's magnitude and phase. After that, the image's mean, standard deviation of the magnitude, and phase values

are calculated. The output is a 4-dimensional feature vector presented as:

$$\begin{aligned} Gabor\_F\_Vectore = [&mean\_magnitude, \\ &mean\_phase, \\ &std\_magnitude, \\ &std\_phase] \end{aligned} \tag{27}$$

- Step 2: The Canny filter generates several edge pixels in each of the several non-overlapping regions of the image. The output is a binary image where one is set for edges and 0 for non-edges. The Canny filter produces a multidimensional feature vector expressed as:

$$\begin{aligned} Canny\_F\_Vector = [&num\_edge1, \\ &num\_edge2, ..., \\ &num\_edgen] \end{aligned} \tag{28}$$

- Step 3: By calculating the gradient in the directions of x and y, the Sobel filter analyses an image separately. The standard deviation and the mean of the gradient direction and the standard deviation and the mean of the gradient magnitude throughout the entire image are calculated by this filter to create a feature vector. The result is a 4-dimensional feature vector, which can be shown just like this:

$$\begin{aligned} Sobel\_F\_Vector = [&mean\_gradient\_magintude\_x, \\ &std\_gradient\_magnitude\_x, \\ &mean\_gradient\_magintude\_y, \\ &std\_gradient\_magnitude\_y] \end{aligned} \tag{29}$$

- Step 4: Feature vectors generated by Canny, Sobel, and Gabor filters are concatenated into a single feature vector. The result is a high-dimensional feature vector that can be used as an input to a random classifier. The concatenated feature vector is represented as:

$$\begin{aligned} F\_vector = [&mean\_gradient\_magintude\_x, \\ &std\_gradient\_magnitude\_x, \\ &mean\_gradient\_magintude\_y, \\ &std\_gradient\_magnitude\_y \\ &num\_edge1, \\ &num\_edge2, ..., \\ &num\_edgen \\ &mean\_magnitude, \\ &mean\_phase, \\ &std\_magnitude, \\ &std\_phase] \end{aligned} \tag{30}$$

For forest image segmentation, it is crucial to preserve raw information by including the original image pixels in the feature vector. Although Gabor, Canny, and Sobel may lose some overall pixel intensity or colour information, they can recover specific patterns (such as texture and edges) [43]. Retaining all visual information important for precise segmentation, including the original pixels, guarantees that the model can access the raw data.

5.2.2 Feature pruning

Features that insignificantly contribute to the segmentation process need to be trimmed off. This study adopted the Principal Component Analysis (PCA) technique to filter only features significantly contributing to image segmentation. The PCA technique effectively reduces the feature vector dimensional by creating a new set of features that have most of the variance of the original image. PCA is an effective dimensionality vector reduction technique that trims irrelevant features, thereby lowering computational overhead [44]. The important aspects are numbered one through

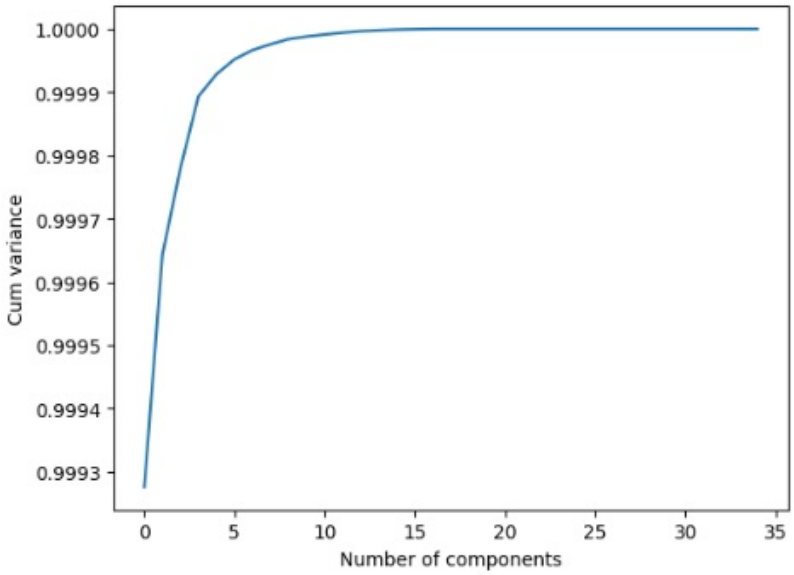

Figure 2: Graph that shows the order of feature importance

fifteen, as seen in FIGURE 2. This indicates that we can recover around 100% of the information inside the first fifteen features; hence, it is safe to disregard the remaining features. To complete the final segmentation process, the 15 characteristics are input into the classifiers like LDA, GNB, LSVM, and RF. The de facto segmentation method in the suggested model will be the classifier that performs the best in the Jaccard Score Index, ROC_AUC, accuracy and RMSE. When the PCA is applied to the first fifteen features, FIGURE 3 shows no correlation between the different obtained principal components. Therefore, we made sure the correlation between the computed PCA was as low as feasible while switching from a feature space of high-dimensional to a feature space lower-dimensional. Thus, the objective of the PCA has been achieved.

5.2.3 Segmentation Using the Classifiers

The data frame's independent variable is the ground truth image pixel value, and its dependent variable is the features produced after the pruning step. X, a vector displayed in Algorithm 1, contains all of the characteristics that were extracted from the original raw image, including pixels and the Gabor, Canny, and Sobel filters. The attributes in question are referred to as independent factors. The ground truth image's pixel values identified as dependent variables are stored in variable M.

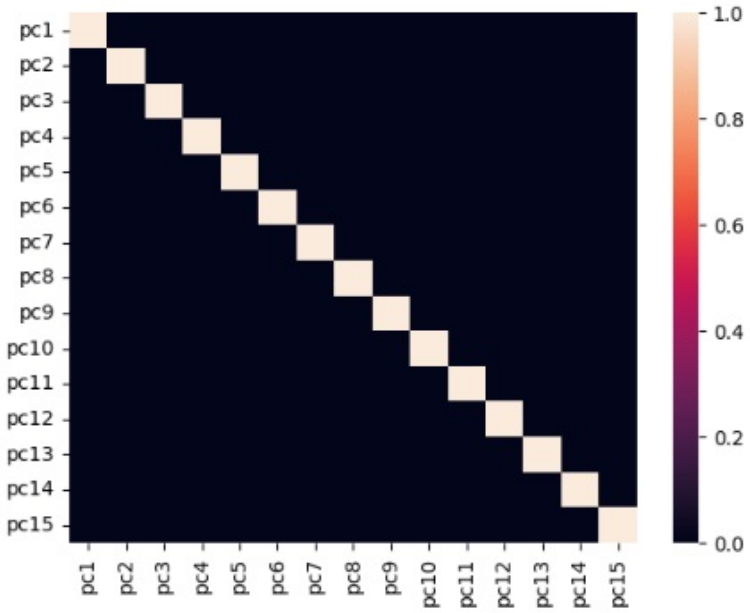

Figure 3: Correlation between features

The random classifier generates predictions regarding the segmented image after separating these two sets into train and test sets. The parameters of the machine learning classifiers were changed, as TABLE 4 demonstrates.

Algorithm 1: Machine Learning Classifier segmentation algorithm

Input: Input: I(b): pixel_values_from_Ground_truth
Input: Input: I(a): pixel_values_from_Independent_variables
for `I(a) = 0` **do**
    $feature1 \leftarrow pixel\_values\_from\_original\_image$
    $feature2 \leftarrow gabor\_filter$
    $feature3 \leftarrow canny\_edge$
    $feature4 \leftarrow sobel$
    $feature5 \leftarrow pixel\_values\_from\_ground\_truth$
end for
$Q \leftarrow \sum_{1}^{4} feature(a)$
$V \leftarrow feature5$
Q ⫫ V
Input: Input : Data: Train_set
Data = train + test
model = MachineLearningClassifier()
model.fit(Q_train, V_train)
prediction_test = model.predict(Q_test)
loaded_model = pickle.load(open(filename, 'RB'))
return *segmented_image*

Table 4: Parameters used in the experiments

Method	Parameters
LSVM	$kernel =' linear', class_weight =' balanced', FeatureScaling = Standardize,$ $param\_grid =' svm\_C' : [0.01, 0.1, 1, 10, 100]$
RF	$n\_estimators = 200, max\_depth = 20, min\_samples\_split = 4,$ $min\_samples\_leaf = 2, max\_features =' sqrt', bootstrap = true,$ $random\_state = 42$
GNB	$var\_smoothing = 1e-9, priors = [0.5, 0.5], random_state = 42$
LDA	$param\_grid = 'solver' : ['svd','lsqr','eigen'],' shrinkage' : ['auto', None],$ $random\_state = 42$

6. RESULTS AND DISCUSSION

This section uses measures like ROC curves, confusion matrix, Jaccard index, Root Mean Square Error, and accuracy to show the outcomes of the suggested model. The confusion matrix provides a tabular structure that shows regions with pixels that have been mistakenly and properly identified, making it easy to visualize misconceptions between regions. The ROC curve, sometimes called a sensitivity measure, is produced by plotting true positives on the y-axis and false positives on the x-axis. A model with a curve that deviates from the diagonal line usually performs better in classification. The ROC curve shows how well the model performs at different threshold values. A model's area enhances its quality. A popular metric for assessing segmentation algorithm predictions is the Jaccard index, often known as the IoU. The intersection of the union area of the segmented image and the reference image defines the IoU or the area of overlap between the predicted segmented image and the ground truth image. The IoU is defined by the formula 31.

$$IoU = \frac{TP}{TP + FP + FN} \tag{31}$$

where the initials FN, FP, and TP stand for truly positive, false negative, and false positive, respectively. The predictions correctly measure the accuracy of the segmentation approach. The accuracy expression can be found in equation 32.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{32}$$

TN symbol stands for the actual negative in equation 32. The square root of the mean square of all mistakes is the error in the square root or root mean square error. For comparing and predicting the errors of various models or model configurations for a single variable, but not across variables, root mean square error is a useful metric because of its scale dependence. One can compute it by using the equation 33.

$$Root\_Mean\_Square\_Error = \sqrt{\frac{1}{n} * \sum_{i=1}^{n} (O_i - P_i)^2}, \tag{33}$$

where the projected value is P_i and the real values are O_i.

As can be seen by the high number of true positives (more than 100,000) in FIGURE 4, the Linear Support Vector Machine findings show that the model did well in predicting positive classifications.

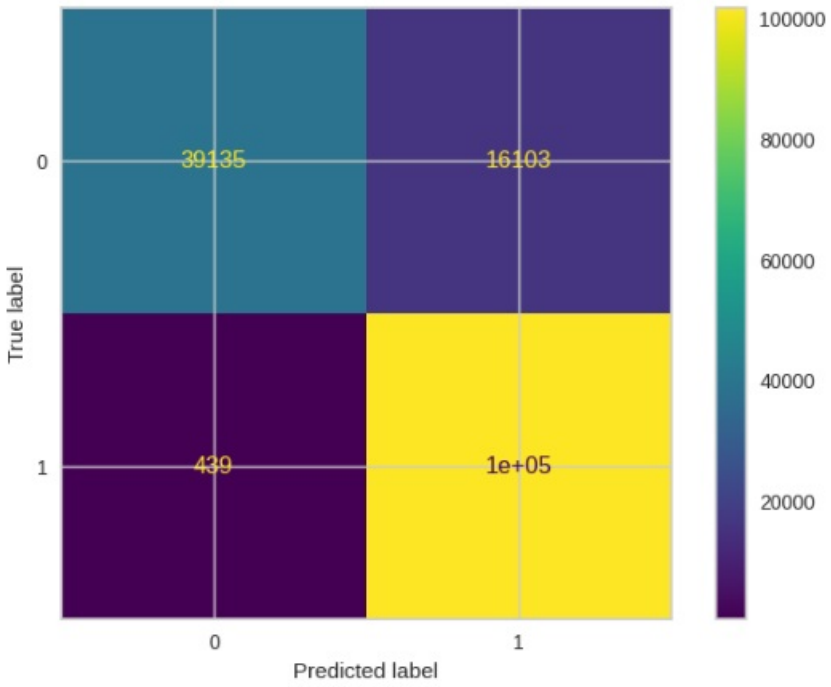

Figure 4: Confusion matrix of linearSVM

However, the model may have trouble distinguishing between the two classes, misclassifying negative cases as positive, as evidenced by the large frequency of false positives (16,103). Due to the low number of false negatives, the model rarely misses true positives (430). FIGURE 5, shows that the ROC curves for both classes are well above the diagonal line, suggesting that they are effective at differentiating between the positive and negative classes. The model's low false positive rate and high true positive rate make it an excellent classifier for most thresholds. With an impressive performance across a range of thresholds, the LinearSVM model has a solid classification capacity, as evidenced by the AUC of 0.94 for both classes.

297 pixels from class 1 were wrongly categorised as class 0, while 16,300 pixels from class 0 were mistakenly classified as class 1, according to the GNB model's confusion matrix (FIGURE 6). Across all classes, the GNB model recognised 55 more pixels wrongly than the Linear Support Vector Machine. The GNB model correctly detected 38938 class 0 pixels and most class 1 pixels, just like a linear support vector machine would. Class distinction is another area where the model excels, as seen by the corresponding ROC_AUC curve for GNB, as shown in FIGURE 7. Yet, compared to the Linear Support Vector Machine, its performance in this area is marginally worse.

The RF-based model outperformed the other three models, as shown by the confusion matrix in FIGURE 8 and the ROC_AUC curve in FIGURE 9. Compared to the other models, the model's misclassification rate was the lowest at 14,762 pixels. Furthermore, as demonstrated by its ROC_AUC value of 0.95, the RF-based model performed better in class differentiation than the other models.

The results indicate that the LDA-based model performed the lowest out of all the models, as indicated by the confusion matrix of Linear Discriminant Analysis in FIGURE 10, and the cor-

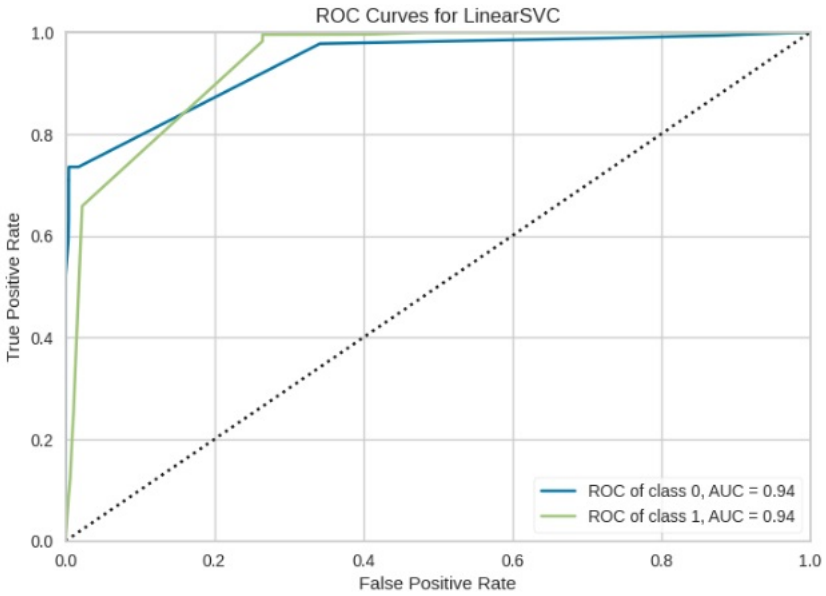

Figure 5: linearSVM ROC_AUC

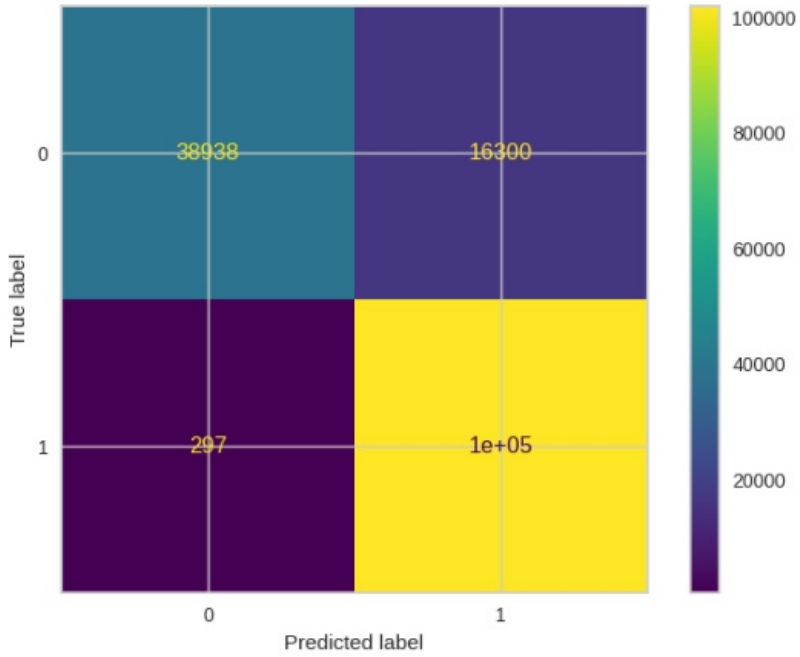

Figure 6: GNB confusion matrix

responding ROC_AUC curve in FIGURE 11. The lowest ROC_AUC value was attained, while the highest pixel misclassification was noted.

Generally speaking, generating false positive results should not be equated with the expense of generating false negative results. ROC curves were used to provide insight into this difficulty.

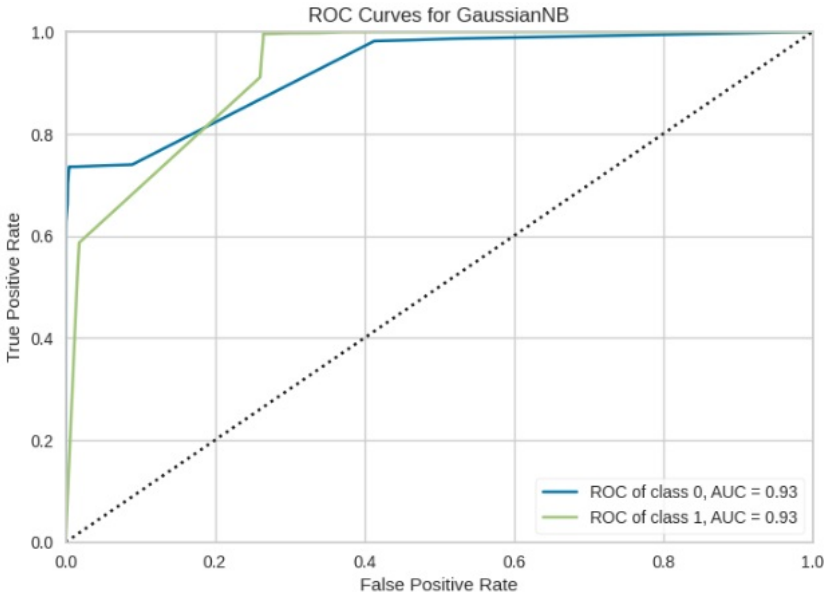

Figure 7: GNB ROC_AUC

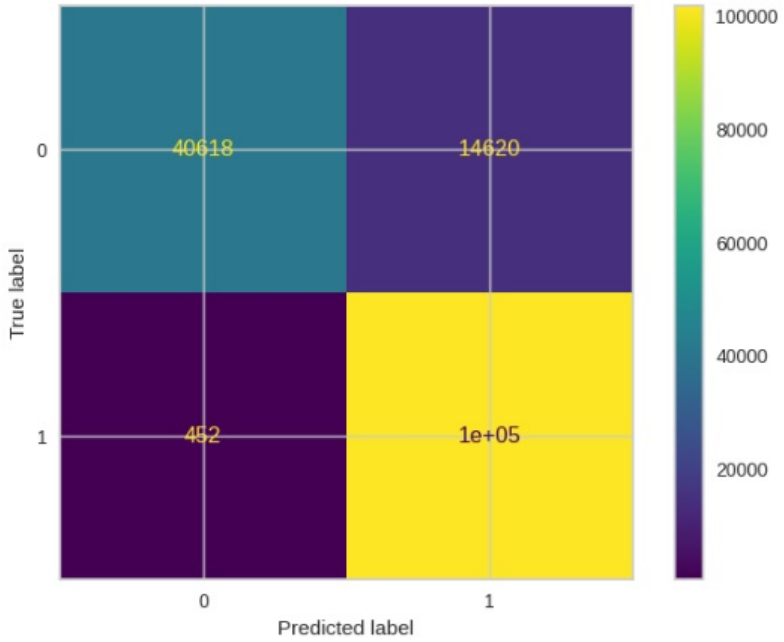

Figure 8: RF Confusion matrix

Applying the specified classification threshold to each category separately shows the ROC curve representing the likelihood of identifying a region or class for a particular model as a function of false positives. If a model has low false positives, it should be able to forecast a region or category well enough to reach the upper left corner of the ROC plot. The GNB incorrectly identified 297 pixels from the forest zone as belonging to a non-forest region based on the confusion matrices. RF

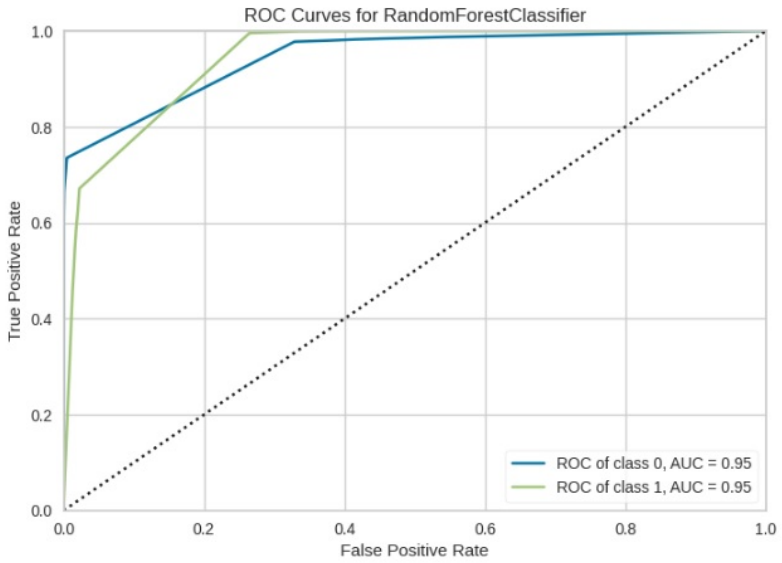

Figure 9: RF ROC_AUC curve

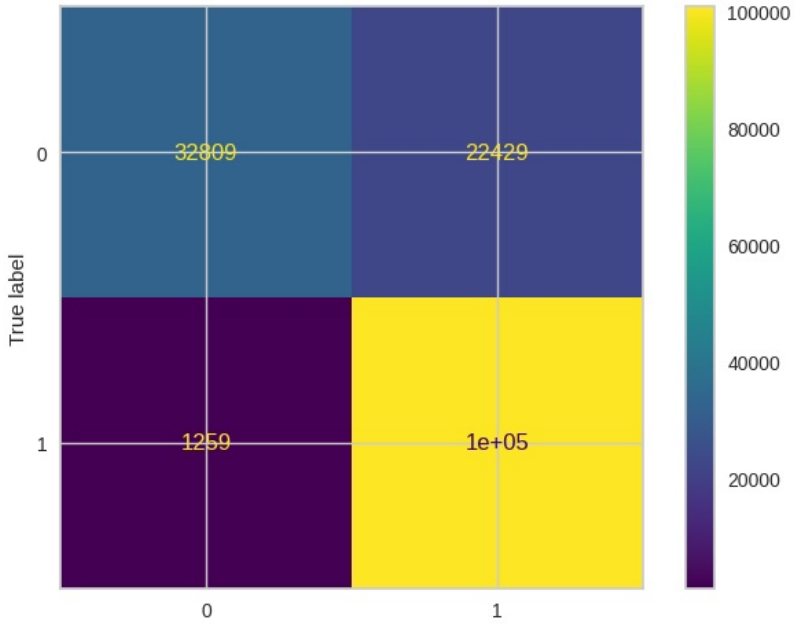

Figure 10: LDA Confusion matrix

(452), LDA (1250), and linear SVM (439) come next. For the fewest (14620) pixels in the forest region, the RF-based model incorrectly identified them as non-forest. Afterwards, LDA (22429), GNB (16300), and linearSVM (16103) come next.

We show in TABLE 5, that according to ROC AUC values, root mean square error, Jaccard score index, and accuracy, our RF-based model performed better than the baseline segmentation tech-

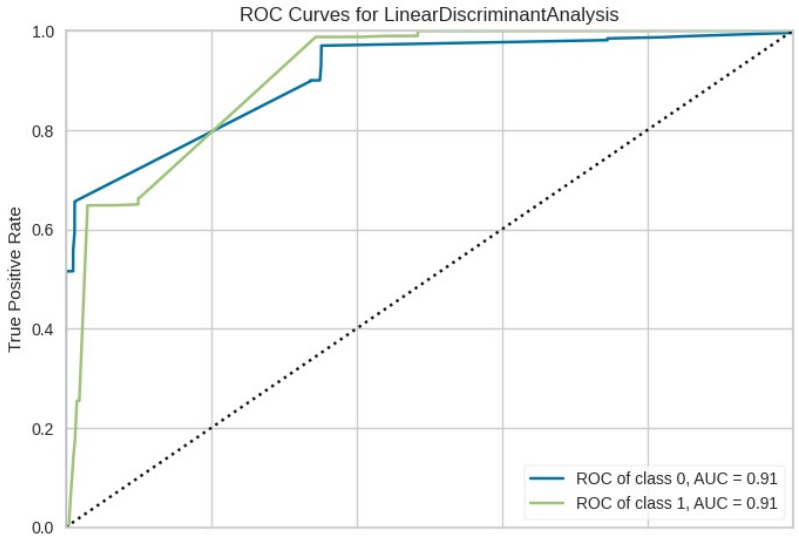

Figure 11: LDA ROC_AUC curve

Table 5: Segmentation results of various methods

Method	Misclassified pixels	ROC_AUC	root mean square error	Accuracy	Jaccard Score
LDA	23688	0.91	0.388	0.849	0.738
GNB	16597	0.93	0.325	0.894	0.809
LinearSVM	16542	0.94	0.324	0.814	0.810
RF	**15072**	**0.95**	**0.310**	**0.904**	**0.825**

niques, including LDA, LSVM, and GNB. Although LinearSVM misclassified 16533 pixels, our model outperforms it with the fewest misclassified pixels (15072). This suggests that the easiest way to distinguish between areas that are in and are not in a forest is with our model. Our RF-based model's predictions are significantly closer to the actual values than those of other models, as seen by its lowest root mean square error of 0.310 for errors. Our suggested model outperformed non-forest region areas in forest regions, as evidenced by its best ROC_AUC value of 0.95. The RF-based model had the highest accuracy of 0.83, indicating that most pixels were correctly sorted into their respective zones, in addition to having the largest IOU of 0.825, which indicates minimal overlap between the target mask and the projected output. The LDA-based model performed better regarding misclassified pixels between the two regions, root mean square error, Jaccard index score, and ROC_AUC values. According to the high ROC_AUC value, our RF-based model is quite effective at distinguishing between forested areas and those that are not. To increase accuracy and achieve the lowest error in terms of root-mean-square error, we offer an RF-based model that predicts pixel values significantly closer to the pixel values of the ground truth image. The segmented areas produced by our suggested model are similar to ground truth images, as seen by the high Jaccard similarity index. The prediction results of the algorithms are shown in FIGURE 12.

Integrating the model with advanced image processing and analysis methods such as object image analysis (OBIA) is necessary to enhance its capacity to recognise and differentiate distinct trees

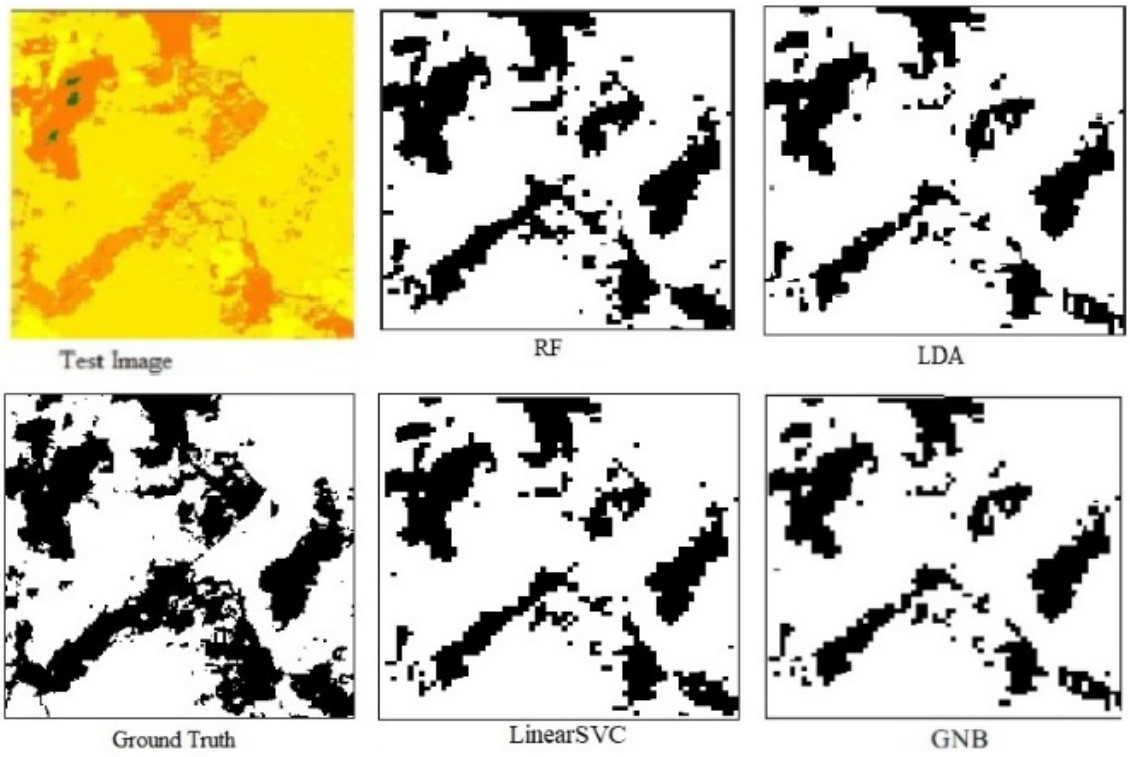

Figure 12: Results of predicted segmentation by RF, LDA, GNB and SVM

based on attributes such as colour, form, texture, and spatial arrangement. Identifying individual trees makes it easier to monitor their development and see how they change over time. Additionally, it offers data on tree height, crown area, and canopy cover.

Table 6: Results from other studies

Technique	**Accuracy**	**IOU**
Spatial pyramid pooling using Unet [45]	86.71	75.59
Hnet with Inception as backbone [46]	68	83
SemisFsNet [47]	-	80
Forest segmentation using Unet[48]	82.55	54
improved tuna swarm optimization (ITSO) [47]	-	59
Unet semantic segmentation [49]	99	97
RF	**90**	**83**

The performance of several algorithms is assessed to identify which segmentation algorithm performs the best. Generally speaking, segmentation performance depends on the kind of images utilised; some perform well when working with remote sensing images but poorly when exposed to synthetic or medical images. Compared to models based on LDA, SVM, and GNB, our model based on an RF approach performed the best in segmenting remote sensing images. Our models outperformed the models in [45–48, 50, 51], as shown in TABLE 6. Nevertheless, our model is outperformed by the unet segmentation in [49], achieving an accuracy of 99% and an IOU of

97%. Unet's capacity to extract high-level and low-level information to carry out the segmentation procedure may be the cause.

7. CONCLUSION

After the Gabor and edge detection filters (such as the Canny and Sobel filters) were used to extract features, the RF was used to divide the aerial satellite images into forest and non-forest sections. The model described in this study proved to be capable of performing the segmentation process with an accuracy of 90% and an IoU of 83%. It's crucial to remember that this model does not address deep learning; it only addresses machine learning. Deep learning techniques use more energy because they are more complicated to train and operate on computers due to their several layers and many parameters. Energy consumption rises during data pre-processing and model training since deep learning models need enormous data to train to function at their best. The ensemble edge vector approach facilitated the model's high effectiveness.

8. FUTURE WORK

As shown in TABLE 5, Random Forest performs well compared to other models from relevant studies and is competitive compared to traditional methods. The best approach is unet semantic segmentation [49], which is especially useful for difficult jobs requiring in-depth feature extraction, as shown in TABLE 6. Future work could incorporate deep learning architectures such as Unet into the suggested model to achieve even better segmentation performance.

Author Contributions

The authors contributed equally.

Funding

This research received no external funding.

Institutional Review Board Statement

Not applicable

Institutional Review Board Statement

Not applicable

Informed Consent Statement

Not applicable

Data Availability statement

The data supporting this study's findings are available on [38]. The authors confirm that the data supporting the findings of this study are available within the article.

References

[1] Körting TS, Garcia Fonseca LM, Câmara G. GeoDMA— Geographic Data Mining Analyst. Comput Geosci. 2013;57:133-145.

[2] Soille P. Morphological Image Analysis: Principles and Applications. Springer. 1999.

[3] Cheng J, Bo Y, Zhu Y, Ji X. A Novel Method for Assessing the Segmentation Quality of High-Spatial Resolution Remote-Sensing Images. Int J Remote Sens. 2014;35:3816-3839.

[4] Soares AR, Körting TS, Fonseca LM, Neves AK. An Unsupervised Segmentation Method for Remote Sensing Imagery Based on Conditional Random Fields. In: IEEE Latin American GRSS & ISPRS Remote Sensing Conference (LAGIRS). IEEE PUBLICATIONS. 2020:1-5.

[5] Gurney CM, Townshend JR, et al. The Use of Contextual Information in the Classification of Remotely Sensed Data. Photogramm Eng Remote Sens. 1983;49:55-64.

[6] Khryashchev V, Pavlov V, Ostrovskaya A, Larionov R. Forest Areas Segmentation on Aerial Images by Deep Learning. In: IEEE East-West Design & Test Symposium (EWDTS). IEEE PUBLICATIONS; 2019:1-5.

[7] Maji S, Malik J. Object Detection Using a Max-Margin Hough Transform. In: IEEE Conference on Computer Vision and Pattern Recognition. IEEE PUBLICATIONS. 2009:1038-1045.

[8] Zhang YJ. A Survey on Evaluation Methods for Image Segmentation. Pattern Recognit. 1996;29:1335-1346.

[9] De Graaf CN, Koster AS, Vincken KL, Viergever MA. Validation of the Interleaved Pyramid for the Segmentation of 3D Vector Images. Pattern Recognit Lett. 1994;15:469-475.

[10] Wang Z, Wang E, Zhu Y. Image Segmentation Evaluation: A Survey of Methods. Artif Intell Rev. 2020;53:5637-5674.

[11] Johnson B, Xie Z. Unsupervised Image Segmentation Evaluation and Refinement Using a Multi-Scale Approach. ISPRS J Photogramm. 2011;66:473-483.

[12] Corcoran P, Winstanley A, Mooney P. Segmentation Performance Evaluation for Object-Based Remotely Sensed Image Analysis. Int J Remote Sens. 2010;31:617-645.

[13] Haralick RM, Shapiro LG. Image Segmentation Techniques. Computer Vision Graphics and Image Processing. 1985;29:100-132.

[14] Cheng J, Bo Y, Zhu Y, Ji X. A Novel Method for Assessing the Segmentation Quality of High-Spatial Resolution Remote-Sensing Images. Int J Remote Sens. 2014;35:3816-3839.

[15] Li Z, Wu XM, Chang SF. Segmentation Using Superpixels: A Bipartite Graph Partitioning Approach. In: IEEE conference on computer vision and pattern recognition. IEEE PUBLICATIONS. 2012:789-796.

[16] Dakshayani V, Locharla GR, Pławiak P, Datti V, Karri C. Design of a Gabor Filter-Based Image Denoising Hardware Model. Electronics. 2022;11:1063.

[17] Lynn ND, Sourav AI, Santoso AJ. Implementation of Real-Time Edge Detection Using Canny and Sobel Algorithms. IOP Conf Ser.: Mater Sci Eng. IOP Publishing.. 2021;1096:012079.

[18] Sonka M, Hlavac V, Boyle R. Image Processing Analysis and Machine Vision. Springer. 2013.

[19] Gonzalez RC. Digital Image Processing. india: Pearson education. 2009.

[20] Pratt WK. Digital image processing: PIKS Scientific inside. Wiley Online Library. 2007:4.

[21] Zhou H, Fu L, Sharma RP, Lei Y, Guo J. A hybrid approach of combining random forest with texture analysis and VDVI for desert vegetation mapping Based on UAV RGB Data. Remote Sens. 2021;13:1891.

[22] Mäyrä J, Keski-Saari S, Kivinen S, et al. Tree Species Classification From Airborne Hyperspectral and Lidar Data Using 3D Convolutional Neural Networks. Remote Sens Environment. 2021;256:112322.

[23] Martínez Prentice R, Villoslada Peciña M, Ward RD, Bergamo TF, Joyce CB, Sepp K. Machine learning classification and accuracy assessment from high-resolution images of coastal wetlands. Remote Sens. 2021;13:3669.

[24] Wang M, Mao D, Wang Y, Xiao X, Xiang H, et. al. Wetland Mapping in East Asia by Two-Stage Object Based Random Forest and Hierarchical Decision Tree Algorithms on Sentinel-1/2 Images. Remote Sens Environ. 2023;297:113793.

[25] Chan AH, Barnes C, Swinfield T, Coomes DA. Monitoring Ash Dieback (Hymenoscyphus Fraxineus) in British Forests Using Hyperspectral Remote Sensing [Hymenoscyphus Fraxineus]. Remote Sens Ecol Conserv. 2021;7:306-320.

[26] He A, He J, Kim R, Like D, Yan A. An Ensemble-Based Approach for Classification of High-Resolution Satellite Imagery of the Amazon Basin. In: IEEE MIT Undergraduate Research Technology Conference (URTC). IEEE PUBLICATIONS. 2017:1-4.

[27] Lmgwal S, Bhatia KK, Singh M. Semantic Segmentation of Landcover for Cropland Mapping and Area Estimation Using Machine Learning Techniques. Data Intelligence. 2023;5:370-387.

[28] Wang H, Hu C, Zhang R, Qian W. Segforest: A Segmentation Model for Remote Sensing Images. Forests. 2023;14:1509.

[29] Chehreh B, Moutinho A, Viegas C. Latest Trends on Tree Classification and Segmentation Using Uav Data—a Review of Agroforestry Applications. Remote Sens. 2023;15:2263.

[30] Drobnjak S, Stojanović M, Djordjević D, Bakrač S, Jovanović J, et. al. Testing a New Ensemble Vegetation Classification Method Based on Deep Learning and Machine Learning Methods Using Aerial Photogrammetric Images. Front Environ Sci. 2022;10:896158.

[31] Guo Q, Zhang J, Guo S, Ye Z, Deng H et. al. Urban Tree Classification Based on Object-Oriented Approach and Random Forest Algorithm Using Unmanned Aerial Vehicle (UAV) Multispectral Imagery. Remote Sens. 2022;14:3885.

[32] Belgacem R, Malek I, Jabri I, et al. Applying a Set of Gabor Filter to 2nd-Retinal Fundus Image to Detect the Optic Nerve Head (ONH). Ann Med Health Sci Res. 2018;8:48-58.

[33] https://inc.ucsd.edu/mplab/75/media/gabor.pdf

[34] Ali M, Clausi D. Using the Canny Edge Detector for Feature Extraction and Enhancement of Remote Sensing Images. Proc IEEE. 2001;5:2298-2300.

[35] Ahmed AS. Comparative Study Among Sobel, Prewitt and Canny Edge Detection Operators Used in Image Processing. J Theor Appl Inf Technol. 2018;96:6517-6525.

[36] Kwenda C, Gwetu MV, Fonou-Dombeu JV. Hybrid Learning Model for Satellite Forest Image Segmentation. In: International Conference on Artificial Intelligence and Soft Computing. Springer; 2023:37-47.

[37] Kwenda C, Gwetu M, Fonou-Dombeu JV. Hybridizing Deep Neural Networks and Machine Learning Models for Aerial Satellite Forest Image Segmentation. J Imaging. 2024;10:132.

[38] https://www.kaggle.com/datasets/quadeer15sh/augmented-forest-segmentation

[39] Kamarainen JK, Kyrki V, Kälvïainen H. Robustness of Gabor Feature Parameter Selection. In: MVA; 2002:132-135.

[40] Nagasankar T, Ankaryarkanni B. Performance Analysis of Edge Detection Algorithms on Various Image Types. Indian J Sci Technol. 2016;9:1-7.

[41] Akar Ö, Güngör O. Classification of Multispectral Images Using Random Forest Algo-Rhythm. J Geod Geoinf. 2012;1:105-112.

[42] Shahana K, Ghosh S, Jeganathan C. A Survey of Particle Swarm Optimization and Random Forest Based Land Cover Classification. In: International Conference on Computing,Communication and Automation (ICCCA). IEEE PUBLICATIONS. 2016:241-245.

[43] Garcia-Lamont F, Cervantes J, López A, Rodriguez L. Segmentation of Images by Color Features: A Survey. Neurocomputing. 2018;292:1-27.

[44] Radzuan NR, Jaafar H, Zabani FN. An Application of Principal Component Analysis in Aspergillus Species Identification. In: 10th Conference on Systems,Process & Control (ICSPC). IEEE PUBLICATIONS; 2022:296-301.

[45] Ru FX, Zulkifley MA, Abdani SR, Spraggon M. Forest Segmentation With Spatial Pyramid Pooling Modules: A Surveillance System Based on Satellite Images. Forests. 2023;14:405.

[46] Umar M, Babu Saheer L, Zarrin J. Forest Terrain Identification Using Semantic Segmentation on Uav Images. 2021.

[47] Wang J, Fan X, Yang X, Tjahjadi T, Wang Y. Semi-supervised Learning for Forest Fire Segmentation Using Uav Imagery. Forests. 2022;13:1573.

[48] Filatov D, Yar GN. Forest and Water Bodies Segmentation Through Satellite Images Using U-Net. 2022. ArXiv preprint: https://arxiv.org/pdf/2207.11222

[49] Pyo J, Han Kj, Cho Y, Kim D, Jin D. Generalization of U-Net Semantic Segmentation for Forest Change Detection in South Korea Using Airborne Imagery. Forests. 2022;13:2170.

[50] Shi L, Wang G, Mo L, Yi X, Wu X et. al. Automatic Segmentation of Standing Trees From Forest Images Based on Deep Learning. Sensors. 2022;22:6663.

[51] Wang J, Zhu L, Wu B, Ryspayev A. Forestry Canopy Image Segmentation Based on Improved Tuna Swarm Optimization. Forests. 2022;13:1746.