Object Classification by Effective Segmentation of Tree Canopy Using U-Net Model

S. Vasavi *, Atluri Lakshmi Likhitha, Veeranki Sai Premchand, and Jampa Yasaswini

Department of Computer Science and Engineering, Velagapudi Ramakrishna Siddhartha Engineering College, Vijayawada, India

Email: vasavi.movva@gmail.com (V.S.); a.lakshmilikhitha@gmail.com (A.L.L.); 208w1a0556@vrsec.ac.in (V.S.P.); 208w1a0525@vrsec.ac.in (J.Y.)

*Corresponding author

***Abstract*—According to the Forest Survey of India and yearly report, Kerala has a total area covered by trees of 2951 sq km, which is 7.59% of the state's total area. In regions like Kerala, identifying objects from Very High-Resolution Satellite (VHRS) images has become a major challenge. Using a method known as "tree canopy", which refers to the area shaded by trees, objects that are covered by trees can be identified. For mapping tree crowns Mask Region-based Convolutional Neural Network (R-CNN) is used. It struggles for accurate segmentation and distinguish objects from the surrounding trees, leading to misclassification and incorrect object masks. It can also be computationally expensive, making it challenging to process high-resolution images in real-time. A deep learning model that uses a semantic segmentation approach is proposed to detect tree canopy covering objects. Dataset with 0.5 m resolution images is prepared from SAS Planet images. The input image is now preprocessed using pre-processing techniques and trained with U-Net. Further, the images are closed using morphological operation to detect the object. The model is evaluated for 25 epochs with an accuracy of 92%. Finally, the objects are classified based on semantic segmentation using U-Net with backbone of ResNet34. The objects are classified as buildings, roads, water bodies and got accuracy of 84%.**

***Keywords*—tree canopy, Very High-Resolution Satellite (VHRS) images, object detection, U-Net, semantic segmentation, ResNet34, object classification**

I. Introduction

Object detection from aerial imagery is a critical task for many applications, including urban planning, disaster response and environmental monitoring. However, accurate detection is often hindered by the presence of tree canopy, which can obscure objects and create false positives. Deep learning models such as U-Net have recently shown great promise in addressing this problem by effectively separating tree canopy from object structures. Finding instances of each object in digital images or real-world scenarios, separating them, analyzing their necessary features for in-the-moment predictions are the main goals of object detection. By leveraging the power of the U-Net model, this paper aimed to achieve accurate and efficient tree canopy segmentation.

To detect the objects that are occluded by tree canopy, U-Net model is used [1]. The model is trained and tested using labeled and unlabelled data. The performance of the model is observed. For training and testing, Kerala dataset is considered. The performance of the model is evaluated and validated on Chandigarh dataset.

Once the object is detected, the next step is to classify the object as buildings, water bodies and roads and is carried out using U-Net with backbone of ResNet34 [2].

A. Convolution Neural Network

Convolutional Neural Networks (CNNs) are a type of deep learning technique which is particularly good at processing and recognizing images. Among the layers that make up this structure are Convolutional layers, pooling layers, and completely connected layers. An Input, Output and numerous hidden layers are assigned with weights for each node to link and make a Convolution Neural Network or ConvNet. This CNN is mostly used in classification, where various features are recognized in an input image are given weights, and these weights are then passed through multiple layers of filters until they reach the output layer, where they are classified into class labels [3].

B. U-Net

It is a network architecture comprising encoder and decoder blocks that is shaped like a "U". It has four encoders and four decoders connected by a bridge. The contracting path at each encoder block results in a doubling of the filter count and the reduction of the spatial dimensions. In the decoder block, spatial dimensions are doubled. It is a fully convolution network designed to gain knowledge with fewer training examples. U-Net is a semantic segmentation technique, describes the process of associating every pixel of an image with a class label [4].

Manuscript received June 6, 2023; revised July 30, 2023; accepted September 4, 2023; published March 26, 2024.

doi: 10.12720/jait.15.3.422-434

C. Tree Canopy

Tree canopy is defined as the uppermost layer of leaves, branches and stems of trees and other vegetation that covers the ground. The canopy layer is crucial for the health of the forest ecosystem as it provides a variety of ecosystem services such as regulating temperature, storing carbon and supporting biodiversity. The canopy also plays a crucial in reducing the effects of the urban heat and also effects air and noise pollution. Tree canopy cover can be quantified using various methods such as remote sensing, ground-based surveys and aerial photography. Tree canopy cover is also an important indicator of the overall health of the urban environment and many cities have launched initiatives to increase the canopy cover to enhance the livability and sustainability of the urban landscape. Hence, tree canopy is a critical component of forest and urban ecosystems and understanding its structure and function is essential for effective management and conservation [5].

D. Very High-Resolution Satellite (VHRS) Images

VHRS Images are known as Very High-Resolution Satellite images which are the snapshots taken by satellites that offer incredibly fine views of the surface of the Earth. Each pixel in this image often corresponds to a very small region on the ground because the ground resolution of these images is typically less than one meter. Applications that require a great deal of information about the Earth's surface, such as mapping urban planning, environmental monitoring, disaster response and other areas, frequently use VHRS imagery.

E. ResNet

ResNet34 is a 34-layer CNN architecture based on the concept of residual learning. Despite the use of skip connections, it enables quick and accurate learning by avoiding intermediary layers. Because of its deep representations, ResNet34 is specifically made for tasks like image classification, object identification and image segmentation. It can extract complex visual data from images. It is a powerful feature extractor because it has already been pre-trained on massive datasets like ImageNet. ResNet34 achieves a balance between model complexity and performance while having a shallower depth than other ResNet variations, making it computationally efficient while still achieving competitive accuracy on a wide range of visual identification applications [6].

F. Motivation

Accurately detecting and classifying objects in complex scenes is a challenging task as tree canopy often obstructs parts of the objects. Traditional methods for object detection and classification struggle to effectively separate the tree canopy from the objects, leading to lower accuracy. The motivation behind the work is to leverage the capabilities of deep learning algorithms to effectively address this challenge by training models to identify and separate tree canopy from objects, improving the accuracy of object detection and classification in complex scenes.

G. Problem Statement

The problem is to demonstrate the effectiveness of U-Net in object detection and classification through the separation of tree canopy. A pipeline of U-Net that leverages, to segment tree canopy from aerial images and then apply object detection techniques to identify objects is required. Further, the objects are classified using Deep Learning Model. The goal is to segment the tree canopy from satellite images, enabling more accurate Identification and delineation of objects. This paper aims to create a reliable and efficient method for object detection and classification that can be used in urban planning, disaster response and other applications.

H. Objectives

- To Create Indian dataset specific to Kerala State that consists of objects occluded with tree canopy.
- To develop a deep learning-based algorithm that can detect and classify objects in complicated scenes accurately.

I. Organization

The outline of the paper is as follows: Section II begins with a review of the literature review on different tree canopy segmentation methods, object detection and classification methods. Section III provides information on the proposed methodology and architecture. The results and conclusion are covered in the fourth and final sections.

II. Literature Review

This section details various literature reviews that are considered as references.

The tree crown delineation is done in several ways [7]. Their method has two steps: Automatic delineation of Individual Tree Crowns (ITCs) and crown separability enhancement. In the first phase, a complex distance map is created using CNN, and the pixels show how far the ITC boundaries are from one another. Then, the crown boundaries and treetop are enhanced on the distance map using a Laplacian Gaussian filter. In the second phase, using a watershed segmentation algorithm, ITCs are automatically identified. Producing an image in which each pixel values represent the separation from the ITC border is the first stage of improving crown separability. ITC delineation has been successfully applied for mapping tree species in forest and urban areas. Finally, using the detected trees as markers, the watershed segmentation technique was applied to LOG-filtered image to produce the delineation of ITC.

- **Advantages:** It is capable of drawing ITC boundaries over four distinct kinds of mangrove forests.
- **Disadvantages:** It limits the impact of opposing factors like understory vegetation and crown clustering.

Coming to the Methodology, Small images are typically taken to detect tree crowns [8]. It is easy to divide a large image into small images. During the pre-processing, the current image is divided into small snapshot. No changes were made to the image's quality, color palette, white balance, contrast adjustment, or shadow reduction in this case. Based on Mask R-CNN model, the backbone architecture and network head are two coupled modules that make up the tree crown segmentation structure. It shows the connection between faster R-CNNs and the Mask. Last but not least, this study combined a deep learning Mask R-CNN based system with images from Google Earth to present a distinctive framework and practical technique to identify individual tree crowns.

- **Advantages**: Their method has good availability and robustness for detecting trees in challenging urban environments.
- **Disadvantages***:* Needs more improvement in the CNN model and optimize the sample design.

The methodology described in [9], with a deep learning strategy, they created an urban tree specific mapping architecture. These images are pre-processed using geometric and atmospheric adjustments. They used Sentinel-2 and Google Earth label images in two stages of these procedures. The UTCD dataset's training and validation sets are used for training CASNet, while the testing set is used to assess the model's initial production. To reconstruct urban tree cover materials throughout a significant portion of the region, the CASNet has been rigorously evaluated with a focus on the study regions for validation. The fundamental element of the suggested approach for generating super-resolution and recognising urban tree coverings is the SPM approach. Then, using the SPM approach, each merged pixel is divided into sub-pixels and given a site for both tree and non-tree labels.

- **Advantages**: It created large-volume UTCD dataset, which helps in further promoting urban tree cover research.
- **Disadvantages***:* It has less training samples on similar tree locations.

For estimation of canopy height using ICESat-2, a point cloud filtering algorithm is used [10]. Here slope is considered as a major role for detecting the canopy height. The canopy height and cross track photon correction resulted in decreased errors because of slope, proving that these techniques lessened canopy height errors. Three different slope types—uphill, downhill, and flatland were taken into consideration. A slope error correction approach is presented in response to their impact on determining canopy height. Their approach is primarily appropriate for photon-counting LiDAR systems in space. Additionally, it enhances the estimation of canopy height's accuracy.

- **Advantages**: ICESat-2 can cover large area and also improves the estimation of canopy.
- **Disadvantages**: In different regions this theoretical method cannot be used for validation.

According to the methodology described in [11], UAV using a LiDAR sensor for calculating canopy height. The point cloud data in photogrammetry is used for obtaining the height. The main benefit of utilizing these approaches is that it provides huge information for larger images when compared to LiDAR. The satellite images are taken for detecting the canopy height. Image resolution should be less than or equal to 1m for better performance. Different RGB spectral and topography characteristics are taken as input to produce height in the maps.

- **Advantages:** Their method is able to predict the height of the vegetation using DN Network. It is less cost and can obtain a higher resolution image.
- **Disadvantages:** It doesn't provide resolution for performance of the model which is less than or equal to 1.

In the methodology reported in [12], UAV image was captured using a Phantom 4-Multispectralcamera.The camera captures an image with visible light of RGB spectrum. Consistent data is required to obtain RGB spectrum. For finding the individual Tree-Crown, accurate position, size and shape is required for obtaining valid data as output. To evaluate the accuracy Mask R-CNN model is used. This model consists of 3 steps. First, an appropriate backbone is required. Second, to evaluate the sample size of the model and third, average pre-epoch time was compared. This study revealed that, in comparison to sample size and sample distribution. When building a model, it is recommended to use RGB spectrum or image.

- **Advantages:** The average pre-epoch time is shorter than random sample. Determination of RGB spectrum is easier than multi-band.
- **Disadvantages***:* The sample size should not range more than 1689 for the measuring of accuracy. The model with multi band is superior compared to RGB.

In the work explained in [13], Using optical and thermal images captured by UAVs with excellent spatial resolution, trees are precisely recovered from analysed urban locations with difficult backgrounds. Based on their brightness and temperature divergence from other thermal imaging elements, trees can be found in remote areas. Combining data from thermal images with the optical image, which has a higher spatial resolution, can help resolve the complex problem. In order to determine how well a DL-model based on thermal data that can be able to distinguish trees, various data approaches are applied first. Finally, the output which is in the form of binary map that has the accuracy which is high and the canny edge detection operator is used for properly identifying trees count, borders, and estimate the size and area of the forest.

- **Advantages**: A U-Net convolution network has an extremely wide dynamic range. Trees in shady areas can be accurately detected by merging visual and thermal images.
- **Disadvantages:** Since there are no fully connected layers in **the** U-Net network, less training data is required.

The methodology in [14] uses a U-Net design for classifying caffeine growing zones with the aid of Sentinel-2 data, making it simpler to monitor annual changes to coffee plantation acreage. A total of 12 U-Net patterns were trained and evaluated. It is close to 95% of the model's performance and value of 0.12 for loss function, the U-Net with the Adadelta optimizer was selected. The approach proved successful in locating the Vietnamese coffee-based ecosystem.

- **Advantage***:* With an accuracy rate of nearly 95%, the U-Net model correctly predicted the area of the caffeine plantation area exporter region of Vietnam.
- **Disadvantage***:* Their model is restricted to Vietnam-based coffee trees.

According to the methodology described in [15], semantic segmentation approach leverages to extract object parts from images. Here, the dataset covers a diverse range of object classes and variations in pose, scale and lighting conditions. For pre-trained semantic segmentation model, U-Net is used to segment objects from input images. Here, the class labels are assigned to each pixel, distinguishing object regions from the background. The objects are extracted based on the segmentation output that is trained on the dataset. The extracted parts are then used for object detection. To reduce duplicate detections, apply post-processing techniques such as non-maximum suppression or bounding box refinement to enhance the object detection results. The performance of the method is compared to other state of art object detection approaches.

- **Advantages**: It provides fine-grained object representation that divides the image into meaningful parts. This helps in improving accuracy of object detection and makes it more robust to variations in object appearance.
- **Disadvantages**: The part-based approach adds an additional layer of complexity to the object detection pipeline and may increase the overall processing time.

In methodology reported in [16] Spectral images from various classes are collected using appropriate sensors or cameras capable of capturing spectral information. These images capture a wide range of wavelengths beyond the visible spectrum. The spectral images are mostly used for classification problems because they contain useful information across the electromagnetic spectrum. Here, for classifying spectral images machine learning algorithms like SVM, k-nearest neighbor and random forest is needed for sophisticated feature extraction of the data. So, in this methodology, a method for classifying spectral images using a Convolutional Neural Network (CNN) approach is proposed. This method involves experimentally acquiring datasets, pre-processing the raw data, designing the CNN, and then analyzing the classification outcomes.

- **Advantages**: They can automatically learn discriminative features from the spectral images without relying on handcrafted feature extraction methods.
- **Disadvantages**: It fails to deal with large datasets because of lack of computational resources and time taken for training.

From methodology [17], the Unmanned Aerial Vehicle (UAV) detection and classification is a crucial task for many military and civilian uses. Convolutional neural networks and UAV RF fingerprints have been used to successfully classify data under conditions of high Signal to Noise Ratio (SNR). This research examines the RF-CNN approach in low SNR settings. First, a spectral entropy drop-based noise-resistant detection approach is presented. After that, without explicitly training CNN for any particular noise levels, it examines how temporal resolution affects classification accuracy. By simply giving priority to time over frequency resolution, tests on public data set showed there is increasing in classification accuracy from 32% to 60% for an SNR of 10 dB.

- **Advantages***:* Deep residual neural networks, such as ResNet, have shown superior performance in image classification tasks, enabling accurate classification on loaded and unloaded UAV images.
- **Disadvantages***:* Fine-tuning a pre-trained model may encounter issues if the domain or distribution of the UAV dataset significantly differs from the original pre-training dataset.

A. Software Requirements

Software Requirements include:

- Operating System: Windows 10 or above.
- Development Environment: Google Colab.
- Python Libraries:numpy, pandas, PIL.
- Deep Learning Frameworks:Tensor Flow or PyTorch.

B. Hardware Requirements

The criteria for the functional platform are RAM:

- 8GB
- Processor: 11th Gen Intel(R) Core (TM) i3-1115G4 @ 3.00GHz 3.00 GHz
- 4 Gigabytes of GPU

III. Materials and Methods

The system architecture, as well as the methods and algorithms used to accomplish the goal, are provided in this section.

A. Architecture

Fig. 1 presents the Mask R-CNN model used for Tree canopy segmentation.

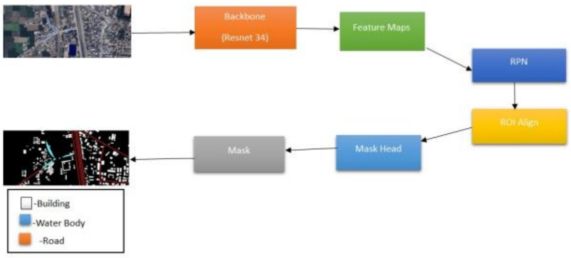


Fig. 1. Tree canopy segmentation system.

Mask R-CNN is used for tree canopy segmentation by training the model specifically on tree-related datasets. Initially, a dataset is prepared that contains 773 images with annotated tree canopies so as to train Mask R-CNN for tree canopy segmentation. The Region Proposal Network (RPN) generates region proposals for potential tree canopies, and the RoI Align and Mask Head branches refine the bounding box coordinates and generate pixel-wise masks for each proposed region. The model is trained using a combination of loss functions, including bounding box regression loss, objectless classification loss, and mask prediction loss. The model processes the input image, generates region proposals, refines the bounding boxes and predicts pixel-wise masks for tree canopies. These masks indicate the precise segmentation of tree canopies in the image. By using Mask R-CNN for tree canopy segmentation, accurate delineation of tree canopies in images is achieved, allowing for various applications such as tree counting, canopy cover estimation and ecological analysis. Table I presents comparison of the existing Mask R-CNN model with the proposed model.

TABLE I. EXISTING MODEL VS. MODIFIED MASK R-CNN MODEL

Parameters	Existing Model [8]	Modified Model
Backbone	ResNet-101	ResNet-34
Input image size	1024×1024	1024×1024
Batch size	2(1)	2(1)
Optimizer	Stochastic gradient descent	Gradient descent
η first phase	0.002	0.005
η second phase	0.0005	0.0005
Epochs	15	25
Minimum detection confidence	0.5	0.6
Anchor scales	32, 64, 128, 256,512	32, 64, 128, 256, 512

Fig. 2 presents the proposed U-Net model with backbone of ResNet-34 used for object detection and classification.

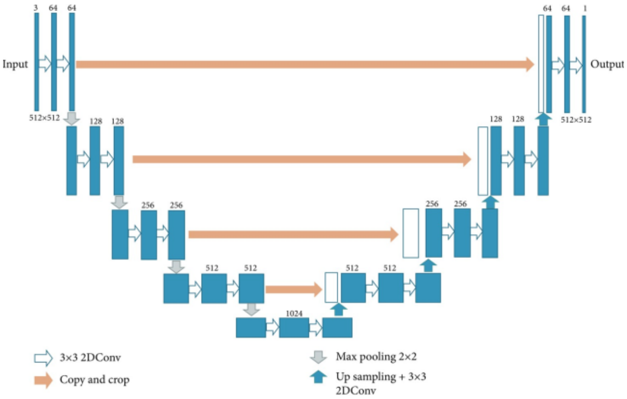


Fig. 2. Proposed U-Net architecture.

In the U-Net model the channels of the input image are buildings, roads and waterbodies with the size of 512×512 the image is processed into the model. By using two strides in the transpose convolution layers, the U-Net achieves an “expansive” or “de-Convolutional” path, which helps recover spatial details and create a dense output with same resolution as input image. A kernel of size 3×3 refers to Convolutional layers use a square filter of size 3 pixels by 3 pixels. As the network progresses, the number of filters increases in encoder to capture more complex features. By considering max pooling with a 2×2 size and a stride of 2, the U-Net architecture progressively reduces spatial resolution of feature maps as the information flows through decoder pathway. This down-sampling helps in capturing larger contextual information while reducing the computational burden. Basically, there are four levels in the encoder and decoder pathway resulting in a total of eight levels. Each level consists of two Convolutional layers, one before and one after the max pooling or transposes convolution operation. This would give a total of 16 Convolutional layers. Additionally, there are skip connections in U-Net, which allow for the fusion of feature maps from encoder pathway with corresponding feature maps in decoder pathway. These skip connections effectively double the number of layers, bringing the total to 32 (16 encoder layers + 16 decoder layers). Finally, the number of channels in output image is 1 resulting whether the object is building or road or water body. Table II presents comparison of existing U-Net model with proposed model.

TABLE II. EXISTING VS. MODIFIED U-NET MODEL

Parameters	Existing Model [2]	Modified Model
Channels in input image	1	3
Shape of input image	(572, 572, 1)	(512, 512, 3)
Strides	1	2
Input Kernel size	3×3	3×3
Initial Number of Filters	64	64
Parameters	1.9M	1.91M
Pooling type	Max Pooling	Max Pooling
Size of Max Pooling at every layer	2×2	2×2
Number of layers	23	32
Number of channels in output image	1	1

B. *Methodology*

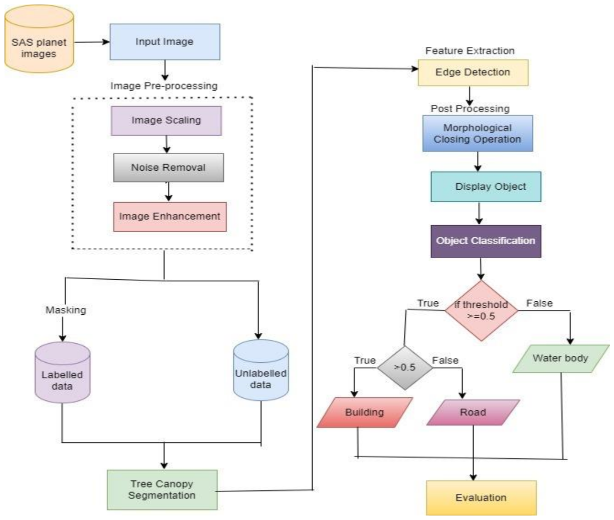


Fig. 3. Process flow diagram.

The paper aims to demonstrate the effectiveness of U-Net in object detection and classification through the separation of tree canopy. The significance of this work lies in its potential to improve object detection and classification accuracy and reduce false positives, which can lead to more effective decision-making in urban planning. As shown in the Fig. 3, the images are collected from the SAS-Planet.

Pre-processing techniques like Image Scaling, Noise Removal and Image Enhancement are done on the images. For Noise Removal, Median Filter is used and Unsharp masking is done for image enhancement. Further, tree canopy segmentation is done by using U-Net. Then feature extraction is performed using canny edge detection method. Post processing of the image is done by using morphological closing operation and finally object is displayed. The next step is to classify the objects as buildings, roads and water bodies based on the threshold value.

C. Algorithms

1) Image scaling

Step 1: Read the input of image of the dimensions 12024×5811 pixels, TIF.

Step 2: Resize the images to get an output image of 512×512.

$$\text{Newwidth}=(\text{Originalwidth})\times(\text{Scalingfactor}) \qquad (1)$$

$$\text{Newheight} = (\text{Originalheight}) \times (\text{Scalingfactor}) \qquad (2)$$

$$\text{Scalingfactor}=(\text{Newsize})/(\text{Originalsize}) \qquad (3)$$

2) Noise removal

The image is then applied Median Filter to reduce noise in the image.

Step 1: Define the size of the neighborhood as $(2k+1)\times(2k+1)$ pixels

Step 2: For each pixel in the image, extract the $k\times k$ neighborhood centered on the pixel.

Step 3: The pixel values are sorted in the neighborhood in lowest to highest order.

Step 4: Now the pixel value is replaced with the typical value of the sorted neighborhood.

Step 5: Repeat the Steps 2–4 for every pixel of image.

The typical value can be calculated using the following formula:

$$\text{Median} = (n + 1) / 2$$

$$\text{Outputpixelvalue}=\text{median}(\text{neighborhoodpixels}) \qquad (4)$$

3) Image enhancement

Un-sharp masking is applied as Image enhancement to improve the standard of the image.

Step 1: Convert the input image to grayscale if it is a color image.

Step 2: Apply a median filter to the grayscale image to create a blurred version. The median filter recovers search pixel value with the typical value of its neighborhood. The size of the neighborhood (kernel size) can be adjusted based on the desired level of blurring.

Step 3: Calculate the high-pass image by removing the blurred image from the original grayscale image.

Step 4: Adjust the intensity of the high-pass image to control the amount of sharpening by multiplying the pixel values of the high-pass image by a sharpening factor.

Step 5: Add the adjusted high-pass image to the original grayscale image to obtain the enhanced image.

Step 6: If desired, convert the final enhanced image back to the original color space.

$$\text{Outputimage} = \text{Originalimage} + \text{Amount}\ (\text{OriginalimageBlurredimage}) \qquad (5)$$

4) Tree canopy segmentation

Step 1: Load the input image dataset and its corresponding ground truth masks.

Step 2: Divide the dataset into training and testing sets

Step 3: Define U-Net architecture

Encoder: A series of Convolutional layers with pooling and ReLU activation functions.

Convolutionformula:conv=Conv2D(filters, kernelsize, activation=′relu′)(inputtensor) (6)

Decoder: A series of Convolutional layers with up-sampling and concatenation of feature maps from the corresponding encoder layer, with ReLU activation function.

Step 4: Train the U-Net model on the training set with a size of 16, 50 epochs and a testing split of 0.2

Step 5: Evaluate the U-Net model on the testing set using the mean Intersection over Union metric.

IoU=intersection (predictedmask, truemask)/ union (predictedmask, truemask) (7)

Step 6: Using the trained U-Net model for predicting the tree canopy and segments the masks for new images.

Step 7: Visualize the predicted segmentation masks on the original images.

5) Edge detection

Step 1: Upload the image to be processed.

Step 2: Transfer the image to grayscale:

$\text{gray} = 0.299 \times R + 0.587 \times G + 0.114 \times B$

Step 3: Apply Gaussian smoothing to reduce noise:

$$G(x,y) = \frac{1}{2} e\left(\frac{(x+y)}{2}\right) I(x,y) \qquad (8)$$

Step 4: Compute the gradient of the smoothed image in both the x and y directions:

$$G_X = \begin{bmatrix} 101 \\ 202 \\ 101 \end{bmatrix} \times G(x,y)\ and\ G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \times G(x,y) \qquad (9)$$

Step 5: Compute the magnitude and orientation of the gradient:

$$M(x,y) = \left(G_x + G_y\right) \, and \; G(x,y) \; = a \tan\left(\frac{G_y}{G_x}\right) \quad (10)$$

Step 6: Apply non-maximum suppression to the gradient magnitude image to thin out edges and reduce noise.

Step 7: Apply hysteresis thresholding to segment the remaining edges into strong and weak edges.

Step 8: Use connectivity analysis to determine which weak edges should be considered as part of strong edges.

Step 9: Extract the resulting edges as features for further analysis.

6) Morphological closing operation

Step 1: Input the image and the desired structuring element.

Step 2: Perform dilation on the input image using the structuring element:

Dilation (Image, Structuring Element) (11)

Step 3: Perform erosion on the dilated image using the same structuring element.

Erosion (Dilation (Image, Structuring Element), Structuring Element) (12)

Step 4: The resulting image is the output

Closing (Image, Structuring Element) = Erosion (Dilation (Image, Structuring Element), Structuring Element) (13)

7) Object detection

Step 1: Provide an image containing objects to be semantically segmented.

Step 2: Load a pre-trained FCN model with a backbone network (such as VGG, ResNet, etc.) that has been trained on a large dataset for feature extraction.

Step 3: Feed the input image through the pre-trained FCN model to obtain a feature map.

Step 4: Use up-sampling layers to transform the feature map back to the original image size.

Step 5: Apply a 3×3 Convolutional layer to generate pixel-wise class predictions.

Step 6: Use a Softmax activation to obtain class probabilities for each pixel, representing the likelihood of each pixel belonging to different object classes.

$$PA = \frac{\sum_{i=0}^{k} p_{ii}}{\sum_{i=0}^{k} \sum_{j=0}^{k} p_{ij}} \quad (14)$$

where, PA = Pixel Accuracy, k = number of types, p_{ii} = true positives (TP), p_{ij} = false positives (FP).

Step 7: The final output is a semantic segmentation mask, where each pixel is assigned a class label based on the highest probability.

8) Object classification

Step 1: Provide an image containing the object to be classified.

Step 2: Pass the input image through the ResNet-34 backbone. The ResNet-34 backbone will consist of Convolutional layers, pooling layers, and residual blocks that automatically learn features from the input image.

Step 3: Add a dense network to reduce features and make the classification predictions.

$$Z = V \times W + b \quad (15)$$

where, V = Vector V, W = weight matrix, b = bias vector and Z = linear transformation.

Step 4: Apply an activation function to the output of the dense network to obtain class probabilities.

$$P(class) = Softmax(Z) \quad (16)$$

Step 5: The final output is the predicted class probabilities for each class.

If the threshold, is less than 0.5 then the object is water body or if it is equal to 0.5 then they are roads or if it is greater than 0.5 then they are buildings and finally the model is evaluated.

9) Evaluation

The evaluation metric is used to calculate the overall execution of the model. The evaluation is calculated on the basis of accuracy, IOU and loss of the model.

$$\text{Accuracy} = \frac{(\text{TP+TN})}{(\text{TP+TN+FP+FN})} \quad (17)$$

$$\text{IOU} = \frac{(\text{TP})}{(\text{TP+FP+FN })} \quad (18)$$

Precision is the fraction of correctly predicted positive samples among all expected positive samples.

$$\text{Precision} = \frac{\text{TP}}{(\text{TP+FP})} \quad (19)$$

Recall is the proportion of all real positive samples that were correctly predicted among all positive samples.

$$\text{Recall} = \frac{\text{TP}}{(\text{TP+FN})} \quad (20)$$

The harmonic mean of recall and precision, which is the F1 score, offers a fair comparison of the two.

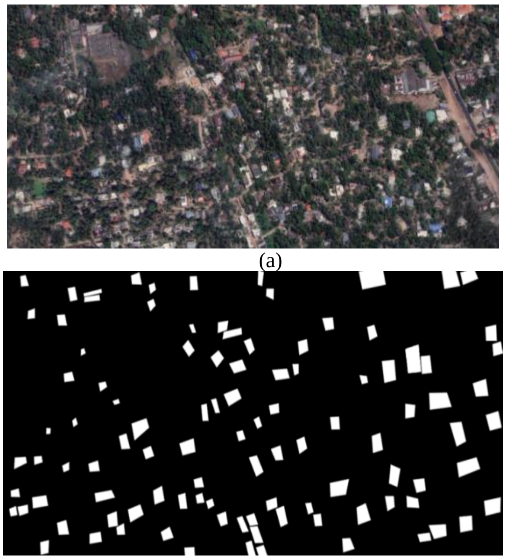
$$\text{F1} - \text{score} = \frac{2 \times \text{precision} \times \text{recall}}{(\text{precision} + \text{recall})} \quad (21)$$

where, TP = True Positive, TN = True Negative, FP = False Positive, FN = False Negative.

IV. Result and Discussion

The outcomes that are obtained from the proposed model have been discussed in this section. Initially, the images are from the SAS-Planet and masked the input images for training and testing the U-Net model. The model has been evaluated with performance metrics like Accuracy, loss, recall, precision and F1-Score.

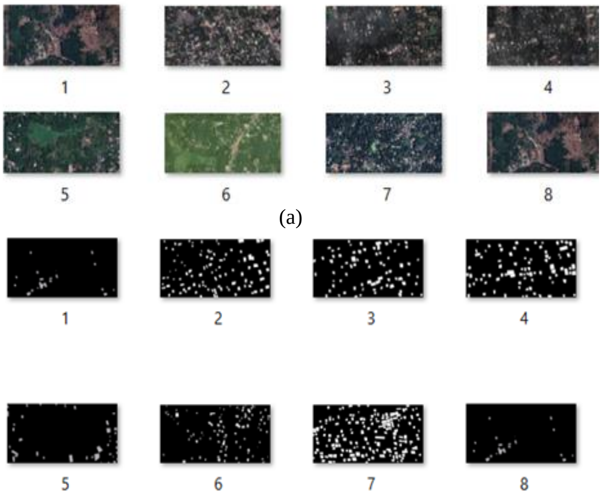
Initially, a sample of 8 images are collected from the SAS-Planet and placed in dataset. Each image size is of 12024×5874 with spatial resolution of 0.5m/pixel in TIF format. The image is then labeled using QGIS and saved with same size that is proportional to the size of the original image. Fig. 4(a) presents a sample input image from the dataset. Fig. 4(b) represents corresponding labeled image.


(a)

(b)

Fig. 4. Sample Input and masked image; (a): Sample input image; (b): Corresponding masked image from dataset.

Fig. 5(a) represents the original images that are collected from the SAS Planet and Fig. 5(b) represents the corresponding labeled images from the dataset. Eight Large images of Kerala are collected from SAS-Planet of size 12024×5874 and Corresponding masks are generated from the satellite images using QGIS of size 512×512.

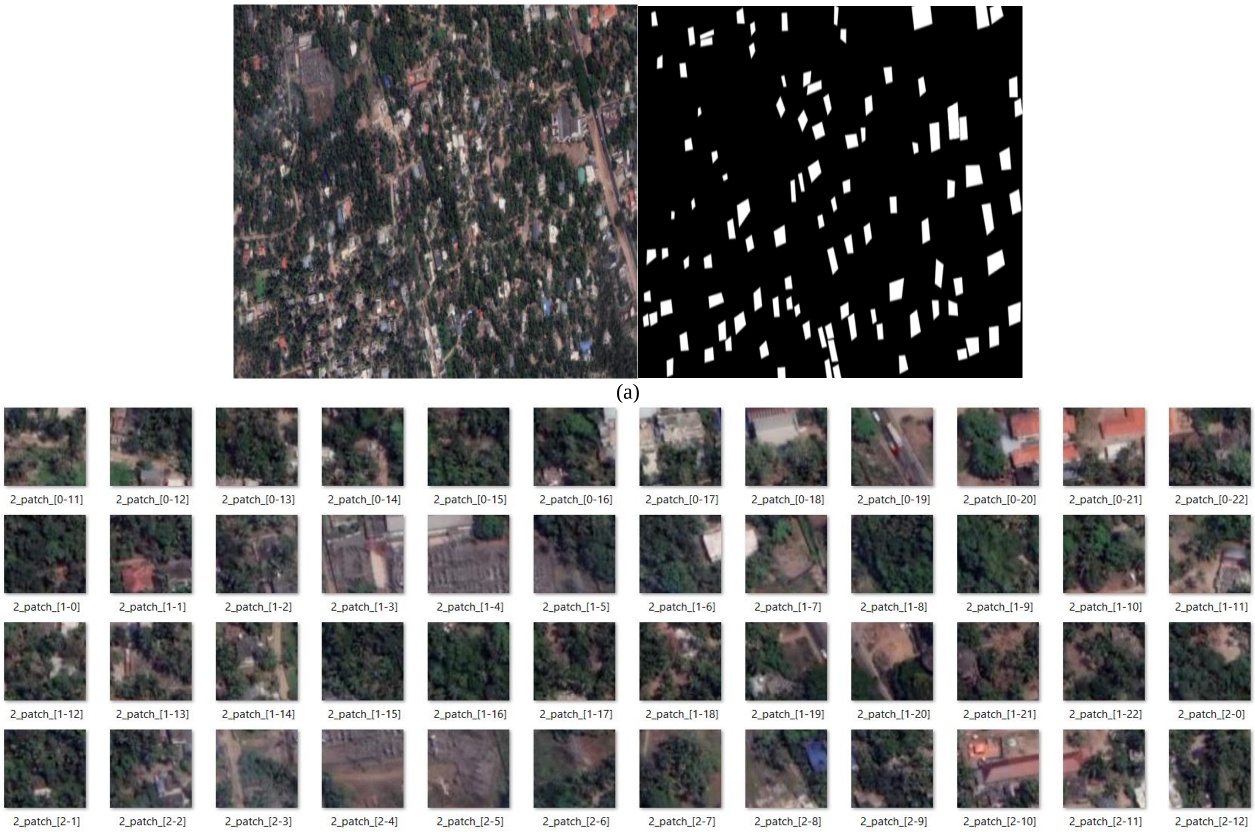


(a)

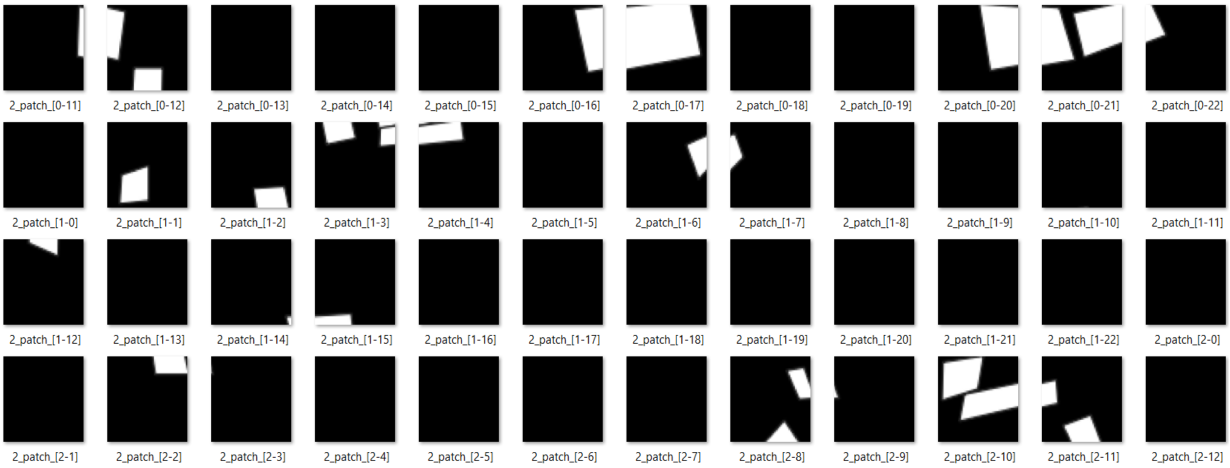
(b)

Fig. 5. Images from the dataset. (a): Original images; (b): Labeled images.

Fig. 6(a) represents the patches of original image and Fig. 6(b) represents the patches of masked image. Each image and their corresponding mask are resized and converted into 253 patches of size 512×512.



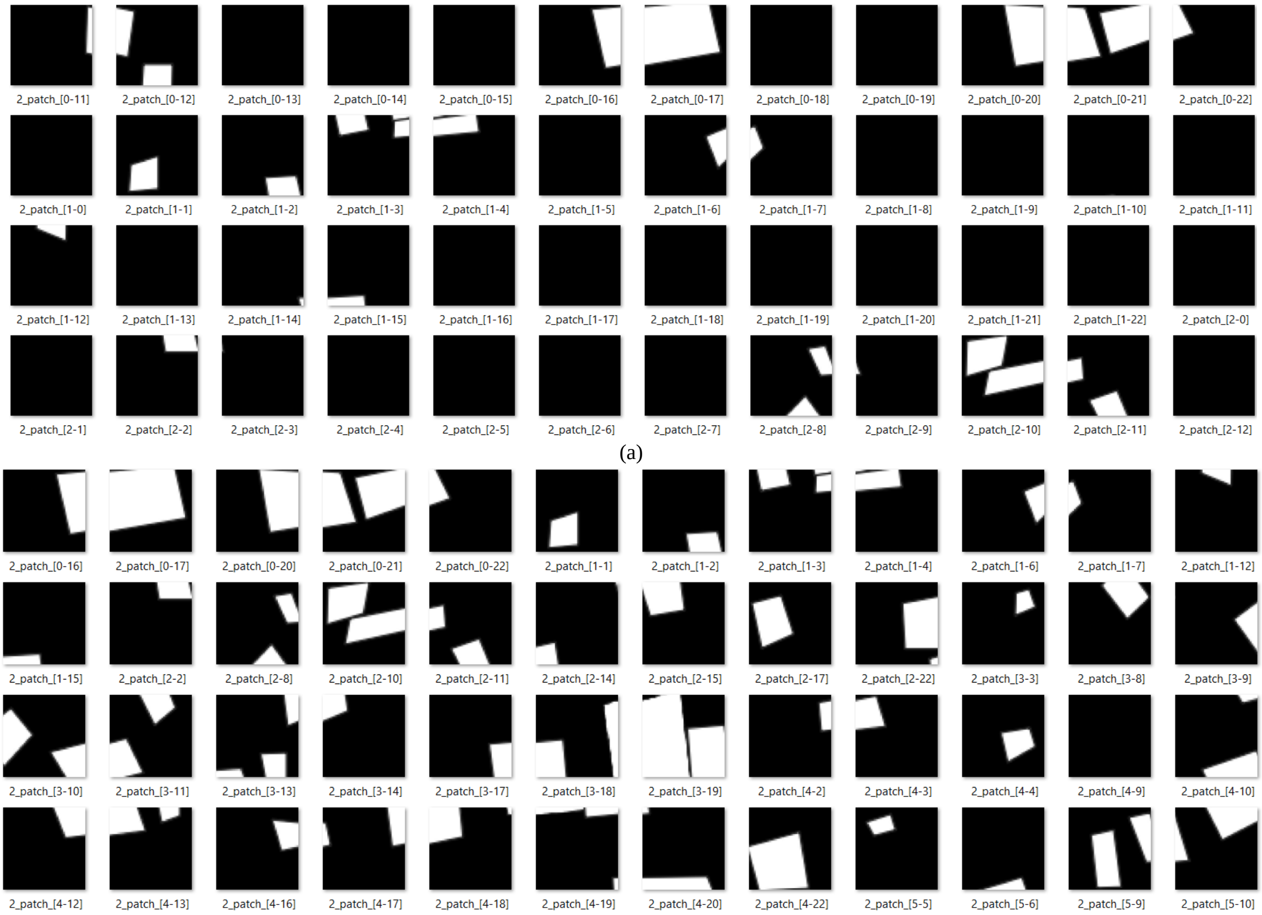
(a)



(b)

Fig. 6. Patched images and masks (a): Patches of image; (b): Patches of masked image.

Fig. 7(a) presents all patched images of an image and Fig. 7(b) presents all the useful images obtained from patched image. The patches consist of 2 types like useful patches and useless patches. The useful patches consist of labeled data while the useless patches do not contain any labeled data or useful information. For training a model only useful patches are considered because the useless patches may affect the performance of the training model.



(a)

(b)

Fig. 7. Finding useful masked images (a): Patched images; (b): Useful images.

The Image is pre-processed using Image Scaling, Noise Removal and Image enhancement. The main objective of pre-processing is to improve the quality of the image so that we can analyze it in a better way. By preprocessing we can suppress undesired distortions and enhance some features which are necessary for the particular application. Resized image is shown in Fig. 8.

Fig. 8. Resized image.

After resizing the image, there will be some noise in the image. Median filter technique is used to reduce the noise present in the image. Median filter is non-linear type of filter [18]. It preserves edges while removing noise by replacing each pixel value with the median of its neighboring pixels. By removing the noise, the image becomes more efficient. Fig. 9 shows the original image and its corresponding median filtered image.

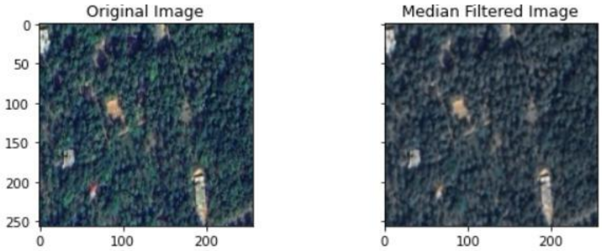


Fig. 9. Original and median filter of the image.

Next step is Image enhancement, it is important because it helps to focus or pick out important features of an image. For enhancing the image, Un-sharp masking is used. For object detection, as it enhances the edges of objects without causing an unrealistic increase in contrast or color saturation. Un-sharp masking technique is used to sharpen an image by subtracting a blurred version of the image of the original image [19]. Fig. 10 represents the Enhanced image.

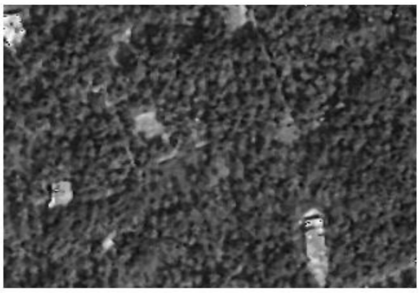


Fig. 10. Enhanced image.

The process of feature extraction is useful to reduce the number of resources needed for processing without losing important or relevant information. In feature extraction, Edge detection is performed using Canny edge detection that provides better edge localization, noise reduction, and thin edge preservation [20]. Additionally, Canny edge detection [21] allows for adjustable thresholds, making it more flexible and adaptable to different image conditions. Results are shown in Fig. 11.

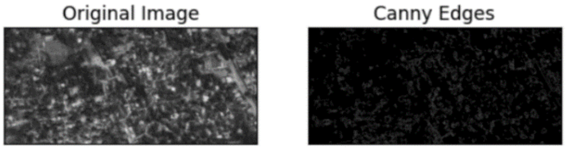


Fig. 11. Original image and edge detection.

The model is trained up to 25 epochs with an accuracy of 92%. Table III compares the accuracy tree canopy segmentation with the existing works.

TABLE III. COMPARISON WITH OTHER MODELS

Study	Methodology	Accuracy Test	Accuracy (%)	Optimizer Used	Dataset
He *et al.* [9]	U-Net Model	ACC	0.91	Adam	Iran
Li *et al.* [10]	U-Net Model	ACC	0.90	Adadelta	Vietnam
Page *et al.* [22]	U-Net Model	ACC	0.82	None	San Joaq-uin
Proposed model	U-Net Model	ACC	0.92	Adam	Kerala

After tree canopy segmentation, morphological operations are used as post processing techniques [23]. By performing a closing operation, we can enhance the object's shape, fill gaps, and improve object detection by creating more complete and connected structures. Fig. 12 represents the morphological closing operation.

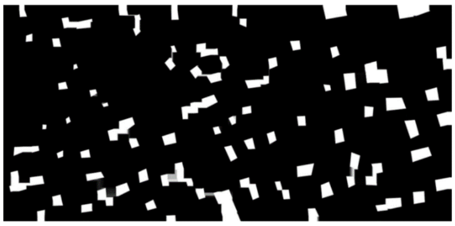


Fig. 12. Morphological closing operation.

Object detection is done using Semantic Segmentation. Each object is now classified into building, water body and road using U-Net with backbone of ResNet-34. Fig. 13 presents object detection for the Kerala dataset. For validating the model Chandigarh dataset is taken.

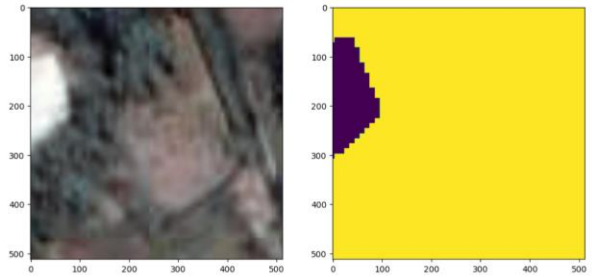


Fig. 13. Detected object.

Performance of object detection is shown in the Figs. 14 and 15. It is observed that the training loss remains constant while the testing loss increased to the peak and then reduced slowly. As epochs increases the testing loss decreases. Performance of object detection is shown in the Figs. 16 and 17. It is observed that the training loss remains constant while the validation loss increased to the peak and then reduced slowly. The validation loss is more during 5 epochs. The validation loss decreases slowly by taking a greater number of epochs. As epochs increases the validation loss decreases. By comparing ground truth bounding box to predicted bounding box, the performance of object is evaluated using Intersection over Union. It is observed that the training IOU increases with the increase in the count of epochs while the validation IoU rapidly decreased and then increases slowly. Finally, at the 25th epoch it reaches maximum 0.65 IoU for validation and for it reaches 0.81 IoU. Here, the model loss is not stable, it took slowly decreases. Table IV presents results of object detection.

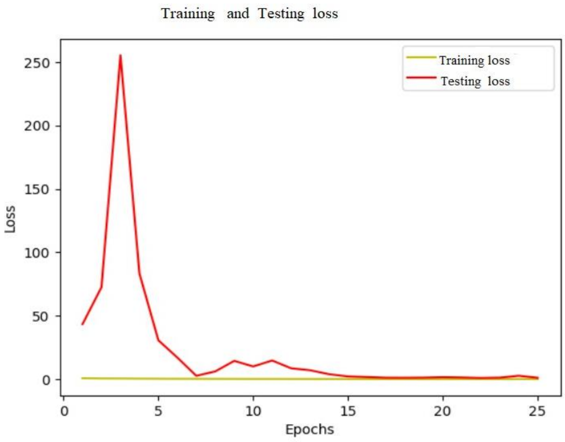


Fig. 14. Loss vs. Epochs graph.

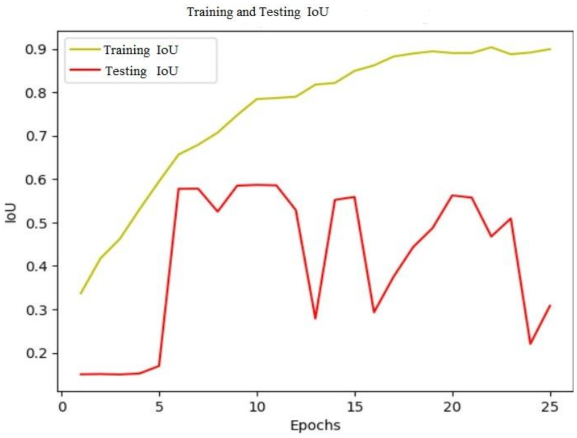


Fig. 15. IoU vs. epochs graph.

The next step is to classify the objects as buildings, roads and water bodies. Fig. 18(a) is the sample input image from Chandigarh dataset, Fig. 18(b) represents the output for classifying the objects. Here, white color represents the buildings, red color represents the roads and blue represents the water bodies.

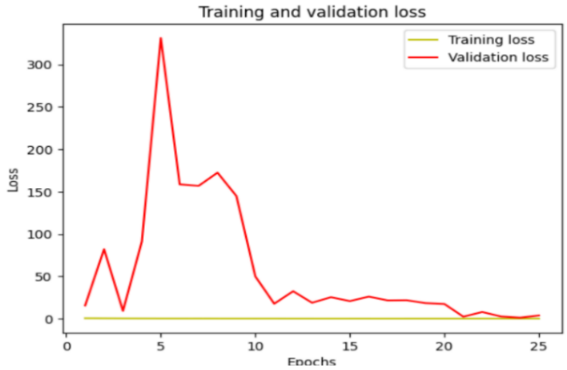


Fig. 16. Loss vs. Epochs graph.

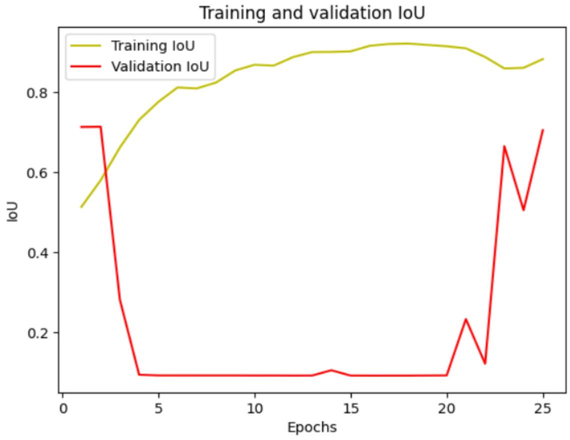
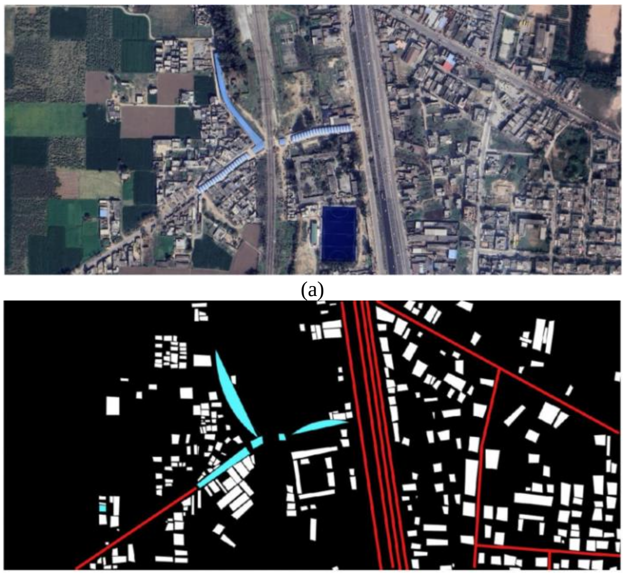


Fig. 17. IoU vs Epochs graph.

TABLE IV. COMPARISON OF TRAINING, TESTING, AND VALIDATION PHASES

Parameter	Training	Testing	Validation
Epochs	25	25	25
Accuracy	0.9593	0.7419	0.8090
Loss	0.1048	1.1940	3.8466
Jaccard coefficient	0.8822	0.5983	0.7047

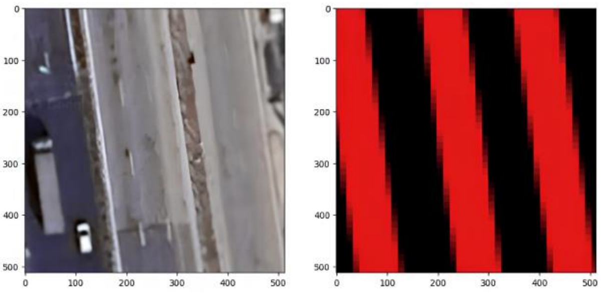


(a)

(b)

Fig. 18. Object Classification: (a) Sample input image; (b) Output for Classifying the objects.

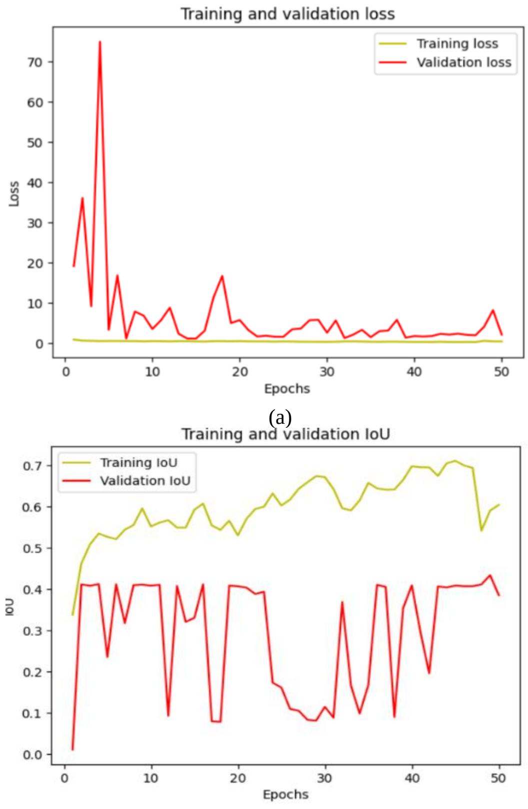
Fig. 19(a) represents the sample input image and Fig. 19(b) shows object detection and classification, roads with red color and displays as the output.



(a) (b)

Fig. 19. Input and classified image (a): Sample input image; (b): Road classification.

The final step is to evaluate the performance of model. The performance of model is assessed for classification of the objects. Fig. 20(a) represents training and validation loss of the model while the Fig. 20(b) represents training and validation IoU vs. epochs graph of model.



(a)

(b)

Fig. 20. Performance graphs (a): Loss vs. epochs; (b): IOU vs. Epochs.

Table V represents the performance evaluation of a model on different datasets taken for testing and validation. The matrix provides valuable insights into the model's accuracy and False Positive Rate (FPR).

$$\text{FPR} = \frac{\text{False Postive}}{\text{False Positive+True Negative}} \tag{22}$$

TABLE V. CONFUSION MATRIX FOR TRAINING, TESTING AND VALIDATION

	Building	Road	Water Body
Building (Training)	398	78	0
Road (Training)	11	95	1
Water Body (Training)	2	6	27
Building (Testing)	96	23	2
Road (Testing)	8	12	1
Water Body (Testing)	0	3	5
Building (Validation)	321	31	1
Road (Validation)	21	82	0
Water Body (Validation)	0	9	25

The accuracy of the model on training dataset is calculated to be 84.14%. This metric indicates the proportion of correctly classified instances out of the total predictions made by the model. An accuracy of 84.14% suggests that the model is performing well and accurately classifying the majority of the objects. Furthermore, the false positive rate is determined to be 16.39%. This rate represents the proportion of misclassified instances where objects were incorrectly predicted. A false positive rate of 16.39% implies that the model sometimes misclassifies objects, leading to false positive predictions.

The accuracy of the model on testing dataset is calculated to be 75.33%. This metric indicates the proportion of correctly classified instances out of the total predictions made by the model. The false positive rate for the testing dataset is determined to be 19.01%. The validation accuracy is about 87.35%. The False Positive Rate is 8.78%.

Table VI represents the comparative study table for the proposed model for training, testing and validation with Precision, recall, F1-Score, and Accuracy.

TABLE VI. COMPARISON OF ACCURACY DURING VARIOUS PHASES

Phase	Precision	Recall	F1-Score	Accuracy
Training	0.960	0.830	0.89	0.84
Testing	0.767	0.750	0.75	0.75
Validation	0.870	0.867	0.86	0.87

V. CONCLUSION

Object detection and classification through effective separation of tree canopy is a complex and ongoing research area that requires interdisciplinary collaboration among computer scientists, remote sensing experts and urban planners among others. These methods can be combined with traditional object detection algorithms to improve their accuracy and reduce false positives and negatives. Dataset is created by taking the images from SAS-Planet. The images are patched, masked and resized into 512×512. The edges of the object are considered as features to be extracted from the image. These images are trained into the model for detection of objects using Kerala dataset. The model is trained for 25 epochs and got accuracy of 92%. Then after the validation is performed, on Chandigarh and got accuracy of 74%. After performing the object detection, the next step is to classify the objects as Building, Roads and Water bodies using Kerala dataset. The classification accuracy is 84% and the validation is performed on Chandigarh dataset and got an accuracy of 87%. The problems that can be faced are High-quality, diverse labeled data is required for training an effective U-Net model for tree canopy segmentation. Rigorous regularization is required to prevent the model from overfitting. Our future work concentrates on improving the accuracy of the model by taking complex occluded scenes.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

S. Vasavi: Conceptualization, Methodology, Writing-Original draft preparation, Validation, Reviewing and Editing; Atluri Lakshmi Likhitha: Software, Writing-Original draft preparation, Validation; Veeranki Sai Premchand and Jampa Yasaswini: Software, Writing-Original draft preparation, Visualization; all authors had approved the final version.

Acknowledgment

The authors wish to thank K. VenugopalaRao, Scientist SG (Retd), Group Director Urban studies NRSC, ISRO, Dr. SC Jayanthi, Scientist SG, Group Head, Urban Studies and Applications (USAG), National Remote Sensing Centre (NRSC), J. Kamini, Sci/Eng. SF HEAD, Urban Studies DIV. NRSC, Reedhi Shukla, Scientist Engineer "SE', Urban Studies Division, NRSC, ISRO, for providing the required data during evaluating the model.

References

[1] T. Hilal and K. T. Chong, "Object detection in very high-resolution aerial images using one-stage densely connected feature pyramid network," *Sensors*, vol. 18, no. 10, 3341, 2008.

[2] G. Cui, "Research on recognition and classification technology based on deep convolutional neural network," in *Proc. 2021 IEEE 3rd Eurasia Conference on IOT, Communication and Engineering (ECICE)*, Yunlin, Taiwan, 2021, pp. 353−357.

[3] N. Darapaneni, B. Krishnamurthy, and A. R. Paduri, "Convolution neural networks: A comparative study for image classification," in *Proc. 2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS)*, RUPNAGAR, India, 2020, pp. 327−332.

[4] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," *Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241, 2015.

[5] J. Martins *et al*., "Segmentation of tree canopies in urban environments using dilated convolutional neural network," in *Proc. 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, Brussels, Belgium, 2021, pp. 6932–6935.

[6] Z. Liu, B. Chen, and A. Zhang, "Building segmentation from satellite imagery using U-Net with ResNet encoder," in *Proc. 2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE)*, 2020, pp. 1967–1971.

[7] G. Lassalle, M. P. Ferreira, L. E. C. L. Rosa, and C. R. D. S. Filho, "Deep learning-based individual tree crown delineation in mangrove forests using very-high-resolution satellite imagery," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 189, 2022.

[8] M. X. Yang *et al*., "Detecting and mapping tree crowns based on convolutional neural network and Google Earth images," *International Journal of Applied Earth Observation and Geoinformation*, vol. 108, 2022.

[9] D. He *et al*., "Generating 2 m fine-scale urban tree cover product over 34 metropolises in China based on deep context-aware sub-pixel mapping network," *International Journal of Applied Earth Observation and Geoinformation*, vol. 106, 2022.

[10] B. Li *et al*., "Correction of terrain effects on forest canopy height estimation using ICESat-2 and high spatial resolution images," *Remote Sens.*, vol. 14, 2022.

[11] S. Illarionova, D. Shadrin, V. Ignatiev, S. Shayakhmetov, A. Trekin, and I. Oseledets, "Estimation of the canopy height model from multispectral satellite imagery with convolutional neural networks," *IEEE Access*, vol. 10, pp. 34116–34132, 2022.

[12] Z. Hao *et al*., "How does sample labeling and distribution affect the accuracy and efficiency of a deep learning model for individual tree-crown detection and delineation," *Remote Sens.*, vol. 14, no. 1561, 2022.

[13] F. Moradi *et al*., "Potential evaluation of visible-thermal UAV image fusion for individual tree detection based on convolutional neural network," *International Journal of Applied Earth Observation and Geoinformation*, vol. 113, 2022.

[14] Q. T. Le *et al*., "Deep learning model development for detecting coffee tree changes based on sentinel-2 imagery in Vietnam," *IEEE Access*, vol. 10, pp. 109097–109107, 2022.

[15] R. Itu and R. Danescu, "Object detection using part based semantic segmentation," in *Proc. 2021 IEEE 17th International Conference on Intelligent Computer Communication and Processing (ICCP)*, 2021, pp. 227–231.

[16] C. López, R. Jácome, H. Garcia, and H. Arguello, "Object classification using spectral images and deep learning," in *Proc. 2020 IEEE Colombian Conference on Applications of Computational Intelligence*, 2020, pp. 1–6.

[17] U. Seidaliyeva, M. Alduraibi, L. Ilipbayeva, and N. Smailov, "Deep residual neural network-based classification of loaded and unloaded UAV images," in *Proc. 2020 Fourth IEEE International Conference on Robotic Computing (IRC), Taichung*, 2020, pp. 465–469.

[18] K. Wu, W. Dong, Y. Cao, X. Wang, and Q. Zhao, "An improved method of median filtering forensics for enhanced image security detection," in *Proc. 2021 International Conference on Networking and Network Applications (NaNA)*, 2021, pp. 308–312.

[19] I. F. Jafar and K. A. Darabkh, "A modified unsharp-masking technique for image contrast enhancement," in *Proc. the Eighth International Multi-Conference on Systems*, 2021, pp. 1–6.

[20] P. Topno and G. Murmu, "An improved edge detection method based on median filter," in *Proc. 2019 Devices for Integrated Circuit (DevIC)*, 2019, pp. 378–381.

[21] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.

[22] D. L. Page, A. F. Koschan, S. R. Sukumar, B. R. Abidi, and M. A. Abidi, "Shape analysis algorithm based on information theory," in *Proc. 2003 International Conference on Image Processing*, Barcelona, Spain, 2003.

[23] P. Salembier, P. Brigger, J. R. Casas, and M. Pardas, "Morphological operators for image and video compression," *IEEE Transactions on Image Processing*, vol. 5, no. 6, pp. 881–898, June 1996.