

COMPARITIVE STUDY OF TREE COUNTING ALGORITHMS IN DENSE AND SPARSE VEGETATIVE REGIONS

Samir Khan^{1,*}, Prasun Kumar Gupta²

¹Computer Science & Engineering Department, H.N.B. Garhwal University, Srinagar Garhwal, Pauri - samirkhaniirs@gmail.com

²Geoinformatics Department, Indian Institute of Remote Sensing, ISRO, Dehradun - prasun@iirs.gov.in

Commission V, SS: Student Session

KEY WORDS: Tree Crown Delineation, Morphological operators, Watershed algorithm, Color space, Convolution Neural Network

ABSTRACT:

Tree counting can be a challenging and time consuming task, especially if done manually. This study proposes and compares three different approaches for automatic detection and counting of trees in different vegetative regions. First approach is to mark extended minima's, extended maxima's along with morphological reconstruction operations on an image for delineation and tree crown segmentation. To separate two touching crowns, a marker controlled watershed algorithm is used. For second approach, the color segmentation method for tree identification is used. Starting with the conversion of an RGB image to HSV color space then filtering, enhancing and thresholding to isolate trees from non-trees elements followed by watershed algorithm to separate touching tree crowns. Third approach involves deep learning method for classification of tree and non-tree, using approximately 2268 positive and 1172 negative samples each. Each segment of an image is then classified and sliding window algorithm is used to locate each tree crown. Experimentation shows that the first approach is well suited for classification of trees in dense vegetation, whereas the second approach is more suitable for detecting trees in sparse vegetation. Deep learning classification accuracy lies in between these two approaches and gave an accuracy of 92% on validation data. The study shows that deep learning can be used as a quick and effective tool to ascertain the count of trees from airborne optical imagery.

1. INTRODUCTION

An automatic tree classification and counting method from aerial imagery can have numerous benefits, such as keeping track of number of tree count in an area which could be beneficial for forest resource management and others. Deforestation is the most arguable issue for every country around the world, therefore, a detailed study of tree count is most required for effective management and quantitative analysis. This study purposes an approach that can automatically segment regions with trees and estimate tree count. However, detecting tree and giving an accurate count can be a tough task, and even inaccurate at times overall result depend on the condition and quality of image taken. This study proposes and compare different approaches for detection and counting trees in a given aerial video.

The morphological reconstruction operations, extended minima & maxima and watershed transformation are few of the widely used techniques used in object detection and segmentation. (Terol-Villalobos *et al.*, 2005); (Shi-Gang *et al.*, 2018). Due to dense packing of tree crowns and due to undergrowth between sparse trees, false positives are a common problem in image segmentation. To remove this ambiguity between plants and trees, a color based segmenting approach was made to discriminate between plants and trees. HSV color space method is well suited for this purpose (Hanbury, 2008) as HSV color removes any illumination in image.

Deep learning has gained a massive popularity over time because of its ability to learn and analyze data at a much faster and accurate way. In recent years researches have reported numerous algorithms for automatic labelling of aerial images in specific categories, many of these include machine learning and deep learning approaches. The result from all these shows that deep learning is the best method over satellite imagery

dataset. Apart from the crown, the aerial images of the tree include many irregularities, unlike manmade structure such as buildings and roads which have definite geometry and are easy to classify.

In order to classify trees through deep learning approach, convolution neural network (CNN) was used for this task. The CNN model is trained with two different datasets having different presence or absence of trees (positive or negative images). The deep learning image classification model is trained in Matlab with parallel computing toolbox for faster processing and acceleration.

This study gives a detail description about using Digital image processing and deep learning methods for delineation and counting trees. Three different approaches are made for tree counting. This paper is organized as follows: in section 2 the background and adoption of existing techniques of all three processes are explained, section 3 explains the methodology for proposed method and in section 4 comparison and discussion of the result from the methods obtained is done.

2. BACKGROUND AND ADOPTION OF EXISTING TECHNIQUES

2.1 Overview of Dense Method Based Detection

In this study, the main approach for tree detection proceed in two main steps. In first step, filtering, segmentation and thresholding operations are applied and in second step Watershed transformation is applied on the image to count the number of trees.

2.1.1 Opening-Closing: The first approach used two morphological operations (1) Opening and Closing, and (2) Opening and Closing by reconstruction.

* Corresponding author

Opening-closing by reconstruction operations is used to remove small features compared to some structural element (SE) geometry, without affecting the shape of object removed. An opening is erosion (shrinking) followed by dilation (regaining the shape) and closing is dilation followed by erosion. Opening by reconstruction is erosion followed by morphological reconstruction. Same goes for closing as well dilation followed by morphological reconstruction. This whole process refines the image by removing small holes and dots that are usually noise. Opening is always followed by closing so as to maintain the original shape of the object, as opening by reconstruction removes small features, and Closing reconstruct those features (Terol-Villalobos *et al.*, 2005).

2.1.2 Watershed Segmentation Algorithm: The watershed transform is a type of region-growing algorithm. The watershed segmentation finds "catchment basins" and "watershed line" in an image by treating it as a surface, where light pixels resemble peaks and dark pixels resemble valleys as shown in figure 1. (Shi-Gang *et al.*, 2018).

Watershed transformation perform segmentation on local regional maxima of the image. In aerial images, there may be more than one intensity maximum. Direct application of the watershed algorithm can cause over-segmentation. To overcome this problem, the tree crown are marked as internal maxima by applying extended minima wherever binary image is non-zero. (Eddins, 2002)

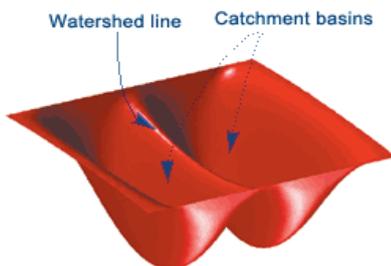


Figure 1 Watershed Diagram (Source: Eddins, 2002)

2.2 Overview of Sparse Method Based Detection

In this study, the main approach for tree detection proceed in two main steps. First applying HSV color space method to filter trees in sparse region and then applying watershed transformation for counting the number of trees.

2.2.1 Hue, Saturation and Value Color Space: The intensity of image pixels is affected by brightness of the features. In order to find a color space which separates the image intensity (luma), from color information (chroma), HSV color model was chosen. The Hue "H" in color model is the portion of the color and is expressed as a number which range from 0-360. Saturation "S" is amount of gray in color it ranges from 0-1 and value "V" is brightness i.e. amount of black or white it ranges from 0-1 (Joblove *et al.*, 1978). The hue value required in this study for trees lies from 40-60. In general, the approach is to segment trees from image just by comparing their hue values with different elements but pixels vary from light to dark shades and with addition of background noise, makes it difficult to identify the perfect range for green pixels of trees.

Test were performed on many different images to detect the perfect range of tree colour pixels. Most of the case tree pixels

hue value lies ranged from greater than 40 to less than 60, hence the trees were separated from the background. However, there remain some noise which could contribute to the tree count like grass, shrubs...etc. which are also filtered out with their respective saturation values.

2.2.2 Watershed Segmentation Algorithm: The process of watershed segmentation is same as discussed in section 2.1.2.

2.2.3 Counting the Trees: At last, the image finally obtained has the tree crowns marked on binarize image and segmented by watershed transformation, which can be automatically counted to get the tree count.

2.3 Overview of Deep Learning Method Based Detection

This method proceeds in two main steps first to train the classifier with labelled images and second step is to implement sliding window algorithm to classify the result and locate candidate tree crown in the aerial video. The tree crown is selected in a way such that the overlapping can be minimized. In this method, only the RGB channel of the images for tree detection and counting is required.

2.3.1 Tree/Non-Tree Labelling: For the first step, that is training a classifier, a training dataset (containing labelled images of trees and non-trees) is required on which the model can be trained. Then a neural network classifier based on visual features classifies the image. Segmented based classifier can capture features that are necessary for the tree identification. The feature selection is done in such a way that a balance is maintained between speed and accuracy.

2.3.2 Deep Learning: Deep learning (also known as deep structure learning or hierarchical learning) is the part of machine learning. Deep learning is an artificial intelligence that acts like a human brain in processing data. The traditional method of programming analyses the data in a linear way. The hierarchical function of deep learning system enables the machine to process data in a nonlinear way for better classification and understanding.

Trend of Deep learning has grown in recent years because of two reasons:

1. Training a network require large amount of data, acquiring data of such extent has recently become possible with development in field of machine learning and big data.
2. Training a network could take from days to weeks, but with the development in graphical capabilities along with cloud computing facilitate training process has reduced from days to hours and weeks to days.

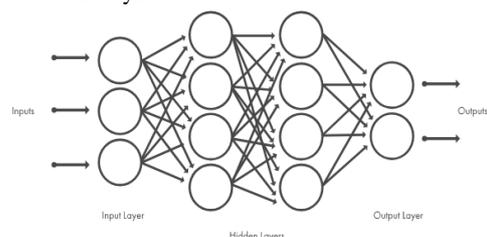


Figure 2 Neural Network Architecture Can Have Hundreds of Hidden Layers (Source: Mathwork.com, 2016)

Figure 2 shows a generalized neural network architecture used in deep learning. A neural network is similar to human biological nervous system, like human brain. It is composed of interconnected neurons which are configured to work in a specific way for pattern recognition or classification.

Deep learning architectures such as deep neural networks have been applied to fields including computer vision, speech recognition, natural language processing, audio recognition, social network filtering, machine translation, bioinformatics, drug design and board game programs where they have produced results comparable to and in some cases superior to human experts.

2.3.3 Convolutional Neural Network: Convolutional neural network (CNN) is a neural network that are made up of neurons that have Weights. Each neuron takes some input, performs some functions on it and optionally follows it with non-linearity. It uses two dimensional convolution layers and is best suited to process the two dimensional data such as images. The manual extraction of features from the network is not required as while training the network, the relevant features are extracted automatically. CNN is best suited for object classification and detection.

The figure 3 shows different layers of CNN and its workflow. Filters are applied and output of each layers serves as input for next layers. The main three type of layers used to make convolution neural network are Convolutional Layer, Pooling layer and Fully-Connected layer. All these layers are joined to make a CNN.

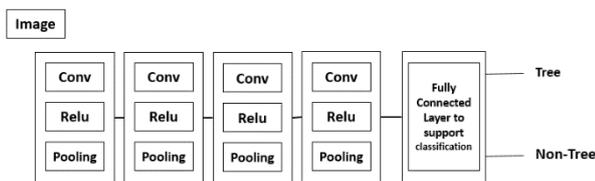


Figure 3 Network with Many Convolution Layers

CNN learns features from images while training by passing through different layers of network, so that they can be used to classify other images. Increasing the number of layers increases the complexity of learning features. Figure 4 shows how a convolution neural network is able to classify object in the image as it passes through different layers of the network.

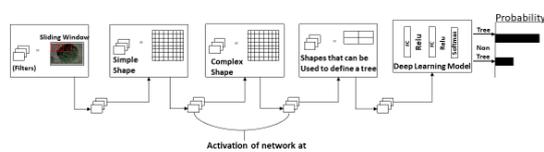


Figure 4 Filters Are Applied To Each Training Image, and the Output of Each Convolved Image Layer Serves As the Input to the Next Layer

2.3.4 Transfer Learning: Transfer Learning in Deep learning is a way of using previously trained network to solve a problem and applying it to solve another related problem by fine tuning. Some of the Pre-Trained network are AlexNet, GoogleNet, VGG-16, etc. Transfer learning is best suited for Category classification as it requires much less data (only thousands are enough rather than millions) and the computational training time is also reduced to minutes instead of hours. Matlab provide tool and function for easy

implementation and designing new models out from these pre-trained networks. (Pratt *et al.*, 1997)

2.3.5 Feature Extraction: Feature extraction is yet another approach for deep learning. Features are learned from training dataset, which is then used to classify the new images and these features can also be used as an input for SVM in machine learning. Since Training can be a bit of a task, as it can take hours to days. Thus, using parallel computing toolbox available in Matlab can ease this task, which uses CUDA code to accelerate the training process using GPUs.

2.3.6 AlexNet: Alexnet is a Convolution neural network, originally written with CUDA to run with GPU support (Krizhevsky *et al.*, 2012). It consists of 11x11, 5x5, 3x3, convolutions, max pooling, dropout, data augmentation, ReLU activations and SGD with momentum layers. It attached ReLU activations after every convolutional and fully-connected layer.

3 METHODOLOGY

3.1 Methodology of Dense Method Based Detection

In this approach, Morphological reconstruction operations (as discussed in Section 2.1) are implemented. The procedure adopted to implement morphology based tree detection is as follows:

1. Read the RGB image and convert the image to grayscale.
2. Use Opening and closing.
3. Followed by opening-by-reconstruction and then closing followed by closing-by-reconstruction.
4. After the opening and closing thresholding is done on the image.
5. Mark extended maxima on the image.
6. Remove over segmentation and mark the extended minima.
7. Apply watershed algorithm to separate two touching tree crowns.
8. Tree counting is done using labelled boundaries.

This process is most well suited for classification of trees in dense region where trees crowns are touching like in evergreen forest or regions having thick vegetation. This method fails to correctly detect trees in sparse region where there is large spacing between trees. Moreover, through experimentation, the study finds that this method works best over region having more than 100 trees.

3.2 Methodology for Sparse Method Based Detection

The steps followed to implement the color space based segmentation (discussed in Section 2.1) is as follows:

1. Normalize image in RGB color space as below (R, G, and B are three color components). Then normalized image is converted to HSV color space.

$$\begin{aligned} R &= R/(R+G+B) \\ B &= B/(R+G+B) \\ G &= G/(R+G+B) \end{aligned} \quad (1)$$

2. This step involves removing background pixels from the image.

$$H1 = \begin{cases} 0, & \text{if } H < b1 \text{ or } H > b2 \\ H, & \text{otherwise} \end{cases} \quad (2)$$

Here b1 is 40 and b2 is 60 (background hue values). Now to remove green grass, though the grass has hue value similar to that of green trees but after experiments the study found that what differ in them is saturation value. So, filtering is done on the bases of saturation value.

$$H2 = \begin{cases} 0, & S \in [S1, S2] \text{ and } V > v \\ H, & \text{otherwise} \end{cases} \quad (3)$$

Here s₁ is the smallest saturation value of the grass/shrubs and s₂ is the biggest one, v is the smallest value of the amount of gray.

After removal of grass and shrubs now the value left is of green trees and rest all are set to zero.

$$G = \begin{cases} 0, & \text{if } H2 == 0 \\ 1, & \text{otherwise} \end{cases} \quad (4)$$

This removes all noise and small dots from the image

3. Segment the processed image by thresholding.
4. Separate touching objects by watershed transformation.
5. Count the tree using labelled boundaries.

3.3 Methodology for Deep Learning Method Based Detection

This study concentrates on the supervised learning method of deep learning for tree classification. The model is trained on the labelled dataset containing images of the positive and negative sample.

3.3.1 Dataset: Tree detection and counting model were trained on dataset consisting of video frames from a synthetic aerial video created by mosaicking true color composited high-resolution satellite images. Frames were extracted from the video using Matlab. The video consists of more than 6,000 frames of different scenes spanning from different regions of south India. The size of each frame is kept at a constant ratio to ease in further processing. The size used in this case was of 540 pixels in width 960 pixels in height, measuring around ~50 kilobyte each. The image used were in RGB color space. Image scenes in the frame covers landscape like rural area, urban area, densely forested, hilly terrain, small to large water bodies, agricultural areas, etc.

Further processing of images is done to fit the criteria for the model. Each individual frame was cropped into 64 x 64 pixels image, hence 128 cropped images of each frame were extracted. After cropping they are classified into two separate classes (1) with trees (positive) and (2) without trees (zero or negative). This tasked was done manually, two separate datasets were created from the cropped image. After creating these datasets, images in each dataset were resized to meet the requirement of the network. Class one consists of 2979 positive images and class two consist of 4228 images of negative sample. Out of these, 2200 images of each class were randomly chosen for training and the remaining were randomly taken as validation data.

3.3.2 Training and Validation of Model: Once the model has been configured, then it is trained in Matlab. First the Training options are set to train the data such as LearningRate, momentum, batchSize, epochs and Optimizer.

Based on these training options the data is then trained. Flowchart of the process is shown in figure 5.

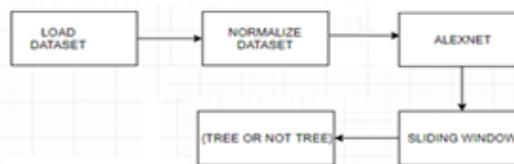


Figure 5 Flowchart of the Process

Before feeding data into training process the data is divided into batches with the finite number of samples in each batch. Batch means a group of training samples. Then the model is trained on these batches, thus ultimately decreasing the training time and loss gradually. In this Model "sgdm" optimizer was used to train the models with a default learning rate of 0.001. The model is trained for 30 epochs.

Validation dataset is used after training the model, to monitor the performance of the model.

3.3.3 Creating Deep Learning Model: The model was created using Alexnet that is deep learning neural network architecture. The model is created and trained in Matlab programming language with the help of neural network Toolbox, Machine learning and statistics toolbox, image processing toolbox and Parallel computing toolbox. The architecture of AlexNet was modified in order to match the requirements. Table1 shows the architecture of the Modified AlexNet:

Table 1 Modified 25x1 Alexnet Architecture

1	'data'	Image Input	227*227*3
2	'Conv1'	Convolution	96 11*11*3 conv stride [4 4] & padding [0 0 0 0]
3	'relu1'	Relu	ReLU
4	'norm1'	Cross Channel Normalization	Cross Channel Normalization with 5 channels
5	'pool1'	Max Polling	3*3 max pooling & stride [2 2] and padding[2 2 2 2]
6	'conv2'	Convolution	256 5*5*256 conv with stride [1 1] and padding [2 2 2 2]
7	'relu2'	Relu	Relu
8	'norm2'	Cross Channel Normalization	Cross Channel Normalization with 5 channels
9	'pool2'	Max Polling	3*3 max pooling with stride [2 2] and padding [0 0 0 0]
10	'conv3'	Convolution	384 3*3*256 conv with stride [1 1] and padding [1 1 1 1]
11	'relu3'	Relu	relu
12	'conv4'	Convolution	384 3*3*192 conv with stride [1 1] and padding [1 1 1 1]
13	'relu4'	Relu	Relu
14	'conv5'	Convolution	256 3*3*192 conv stride [1 1] & padding [1 1 1 1]
15	'relu5'	Relu	Relu
16	'pool5'	Max Polling	3*3 max pooling with stride [2 2] & padding [0 0 0 0]
17	'fc6'	Fully Connected	4096 Fully Connected layer
18	'relu6'	Relu	Relu
19	'drop6'	Dropout	50% Dropout
20	'fc7'	Fully Connected	4096 Fully Connected layer
21	'relu7'	Relu	Relu
22	'drop7'	Dropout	50% Dropout
23	'fc8'	Fully Connected	1000 fully connected layers
24	'prob'	Softmax	Softmax
25	'output'	Classification Output	

4. RESULT

4.1 Result of Estimation of Tree Count on Images

This section discusses the results from the Matlab based implementation of the 2 methods discussed in Section 3.1 and Section 3.2.

4.1.1 Result from Dense Based Method:

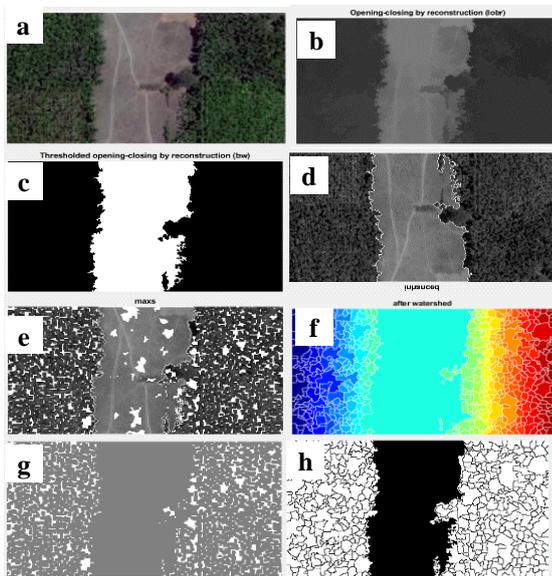


Figure 6 (a) Original image, (b) Opening and Closing by Reconstruction, (c) Thresholded image, (d) Overlay Image, (e) Regional Maxima, (f) Watershed Segmentation, (g) Reduced Regional Max, (h) Watershed Segmentation

Figure 6a shows the input image on which the processing has to be done and estimation of tree count is to be made. The image is then fed to morphological operations where erosion and dilation take place. Figure 6b shows the image after morphological operations with most of the background highlighted. The thresholded image is shown in figure 6c, after morphological processing. Figure 6d shows an overlay on original image and thresholded image with having its perimeter marked. In figure 6e after marking the perimeter the region of interest is selected and local regional maxima are marked on the image. Figure 6f shows the result of watershed segmentation on the image. The result after watershed segmentation is shown in figure 6g, however the image has been oversegmented due to more number of local regional max on the image. In order to avoid oversegmentation the number of regional max are reduced by marking tree top as internal markers and rest all as external markers. The result of which is shown in figure 6h in which the image is almost correctly segmented and oversegmentation is removed.



Figure 7 Output from Dense Based Method

Figure 7 shows the Output as original image with tree count. The number of trees found in the image are 565. By manual interpretation, there are around 637 trees.

4.1.2 Result from Sparse Based Method

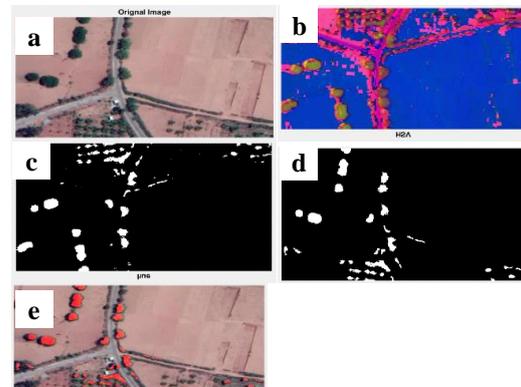


Figure 8 (a) Inputted image, (b) HSV image, (c) Thresholded image, (d) Refined and filtered image, (e) Trees highlighted with orange color

Figure 8a shows the input image on which the processing has to be done and estimation of tree count is to be made. Figure 8b shows the HSV image after conversion. After which filtering is done on HSV image, the process begins with removing background pixels. Secondly, the region which can cause ambiguity with trees counting process are filtered out. Figure 8c shows the thresholded image after filtering. Figure 8d shows the image after selecting the region of interest and now, the regional maxima are marked on the image, and the image is then passed for watershed segmentation. The output of which is shown in figure 8e where trees are highlighted with orange color.



Figure 9 Output from Sparse Based Detection Method

The output is shown in figure 9 with original image and number of trees. The number of trees found in the image is 39. By manual interpretation, there are around 41 trees.

4.1.3 Result from Deep Learning Based Method:

The outputs predicted by the neural network is shown in figure 10(a-d). The total number of trees predicted through deep learning approach are 32, 0, 4 and 845 respectively.

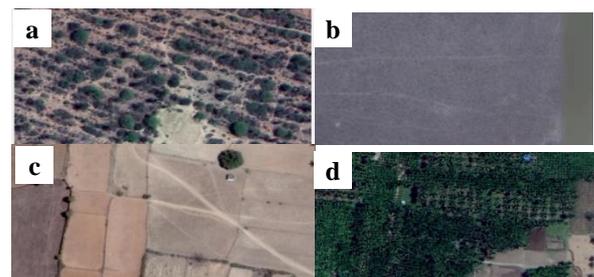


Figure 10 Frames from video with predicted count of (a) 32, (b) 0, (c) 4, and (d) 845 trees

4.2 RESULT ANALYSIS

4.2.1 Result Analysis of Tree Count by Dense Method:

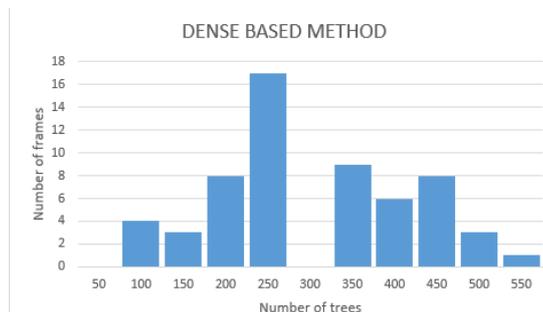


Figure 11 Graph Depicting Number of Trees Vs Number of Frames from Dense Based Method

The figure 11 shows the graph of number of frame vs number of trees counted through dense method on 60 frames.

4.2.2 Result Analysis of Tree Count by Sparse Method

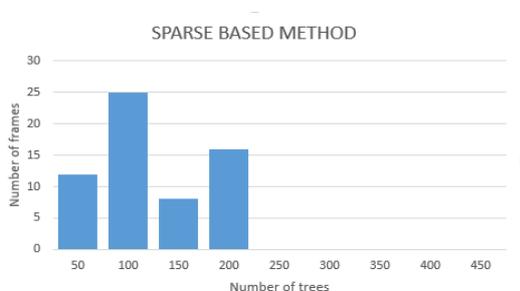


Figure 12 Graph Depicting Number of Frames Vs Number of Trees from Sparse Based Method

The figure 12 showing the graph of number of frames vs number of trees counted through sparse method on 60 frames. It can be clearly seen that the sparse method fails to detect any trees in the frames with higher number of tree count.

4.2.3 Result Graph by Deep Learning Method:

4.2.3.1 Accuracy Graph and Loss Graph:

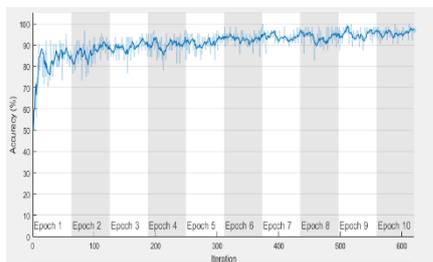


Figure 13 Accuracy Graph

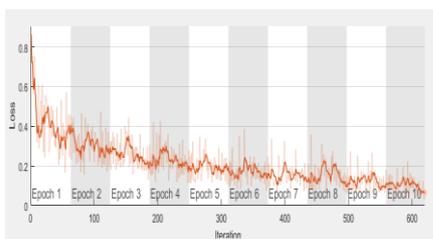


Figure 14 Loss Graph

After training the model was able to get the best accuracy of 92% as the validation accuracy and minimum loss value of 0.024. Figure 13 and figure 14 show the accuracy and loss function graph of the model respectively.

4.2.3.2 Confusion Matrix:

Table 2: Confusion Matrix

True Tree Counts	Not a tree	1997	271
	tree	1	1171
		Not a tree	tree
		Modeled Tree Counts	

Table 2 shows confusion matrix over the test data, from which the study concludes that out of 2268 negative sample only 271 samples were misclassified as to be of tree class. On the other hand, out of 1172 (positive) images only 1 image was misclassified as to of non-tree class (negative image). This shows that the model is very efficient in classifying the tree class from any other class.

4.2.3.3 Analysis of Models:

Table 3 Analysis of Graph

Epoch	Iteration	Time Elapsed (secs)	Test Loss	Test Accuracy	Learning Rate
1	1	2.45	0.826	42.19%	0.001
5	250	645.86	0.167	85.75%	0.001
10	600	1610.93	0.077	88.44%	0.001
15	900	2436.77	0.054	92.88%	0.001
20	1200	3245.60	0.012	96.42%	0.001
25	1500	4410.93	0.024	94.88%	0.001

In Table 3 detail analysis of the model used for detecting and counting tree model is shown. The best model was trained for 25 epochs with base learning rate of 0.001, this model was able to attain 92% accuracy with the validation data when tested.

4.2.3.4 Result Graph of Tree Count by Deep Learning Method

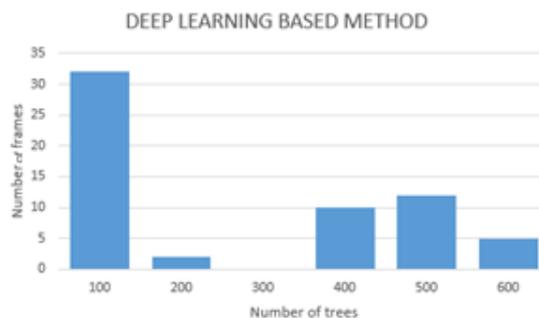


Figure 15 Graph Depicting Number of Frame Vs Number of Trees Counted Through Deep Learning Method on 60 Frames.

The Figure 15 shows the number of frame vs number of trees counted through deep learning method on 60 frames

4.3 Comparative Study

In Figure 16 the resultant graph for dense and sparse algorithm on 550 image frames is shown. Where output from dense algorithm is plotted with thick lines and of sparse algorithm with thin lines whereas, the red dots depict the manual count. By the graph the conclusion was made that the count was accurate in the respected type of vegetated regions.



Figure 16 Graph Depicting Number of Trees Vs Number of Frames

Figure 17 shows the resultant graph of tree counting along with the deep learning (green) dense (orange) and sparse (blue) approach, to get a comparative study between all the approaches made above. The graph is plotted on more than 1600 images. From graph the conclusion was made that the deep learning approach takes the middle path. It also offers the additional advantage of not having to provide the information about density of trees.

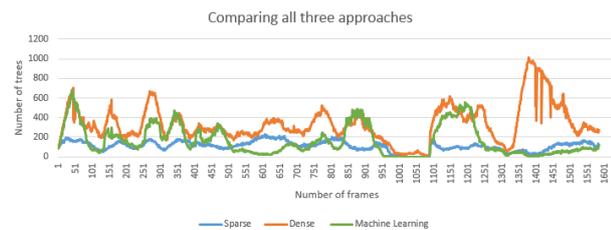


Figure 17 A Comparative Study of All Three Methods Discussed

4.4 Segregation of Images for Application of Dense and Sparse Methods

The Dense based method algorithm is well suited for dense vegetation and perform well with more numbers of trees, whereas the algorithm from sparse method performs accurately with sparse vegetation (region having less number of trees). To merge the algorithm as one, some measures are considered into account such as the area of vegetation. Area of green cover can be a prominent feature for the algorithm to distinguish between dense and sparse vegetation. The limit for this separation is considered on the bases of experimental observations. The area of the region is calculated after the segmenting image. As after the segmentation, only the area on which data is to be processed is left, that is the total area of green vegetation or the Region of interest (ROI). In the segmented image, the area of interest i.e. ROI is marked with the white pixels and the background is marked with the black pixels, so if the white pixels of the image are considered the green area or the tree cover can be classified. On basis of which, the image can be classified to which category it falls.

Figure 18 shows the flow chart for both the algorithm combined and a decision was made on the bases of the area it covers to identify the dense and sparse region, hence a better approach is made for the identification of individual tree in an image.

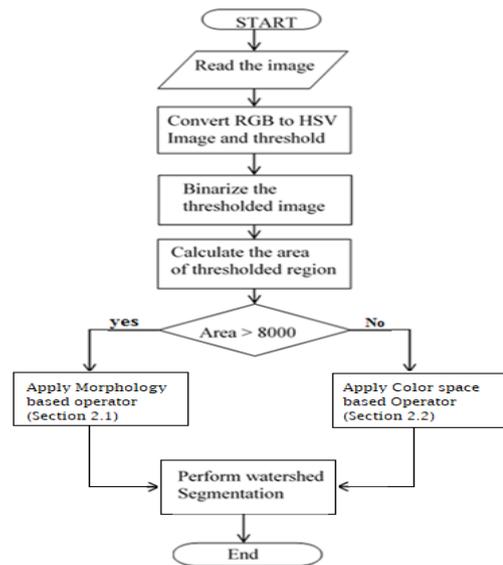


Figure 18 Combined Image Processing Approach for Detection and Counting Trees

4.4.1 Review of Methods in Tree Counting Analysis

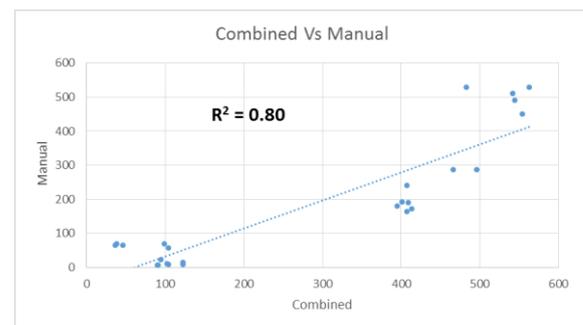


Figure 19 Combined Approach (Dense And Sparse) Count Vs Manual Count.

Figure 19 shows scatter graph of combined algorithm (Dense and sparse) count vs manual count with trending line as manual count. Here R^2 is total variability in dense count that is explained by its regression on manual count.

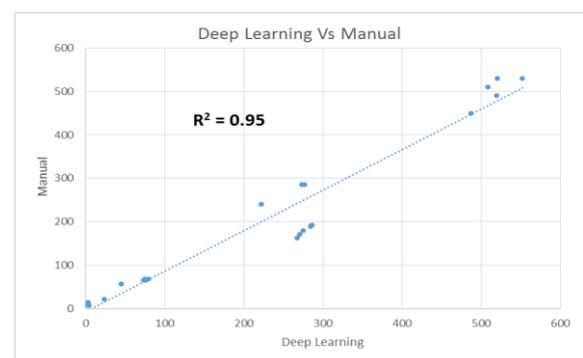


Figure 20 Deep Learning Count with Manual Count

Figure 20 shows scatter graph of deep learning count vs manual count with trending line and R^2 value.

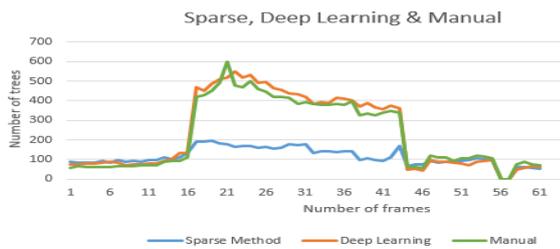


Figure 21 A Comparative Graph Between Sparse Count, Deep Learning Count and Manual Count.

Figure 21 shows line graph comparison of deep learning count, sparse count and manual count. From the graph the conclusion can be made that deep learning count is close to sparse count and manual count in the sparse region.

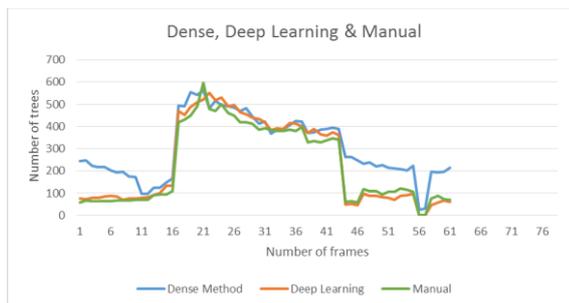


Figure 22 A Comparative Graph Between Dense Count, Deep Learning Count and Manual Count

Figure 22 shows line graph comparison of deep learning, dense count and manual count. From the graph the conclusion can be made that deep learning count is close to dense count and manual count in the dense region.

5 CONCLUSION

A study by researchers has shown that only about 3 trillion trees are left on earth which is 46% lesser than what were there since the start of human civilization (Crowther *et al.*, 2016). Billions of them are being chopped down every year to make houses, roads, buildings etc. The approaches compared in this study proposes to quickly quantify tree loss either by forest fires, urbanization, deforestation or other forms of destruction.

Although tree crown detection and counting can be a challenging task with the development of modern technology and advancement in this field it has become a lot easier. This study, carry out a morphological process for segmenting and watershed transformation to delineate touching crown for the process of counting trees in dense vegetative region.

An HSV color based approach was also implemented for counting trees in the sparse vegetative region where there can be a high chance of getting a false tree count. This approach was successfully able to detect and count trees in the sparse region with an accurate count. The result of this algorithm indicates that it is best suited for counting trees in sparse region. The combined sparse and dense tree counting algorithm gave an R^2 value of 0.80.

For the Deep learning approach, a dataset was prepared, model was trained and used the trained model along with sliding window algorithm to detect trees, different models were

trained which differ in their architectural structure. Dataset was improved further for analysis and training purpose. The pre-trained network AlexNet in Matlab with the help of neural network Toolbox was used for training. After comparing different models in the training process and measuring their performance on the dataset, the best network over training time and accuracy over validation data was chosen. This model was able to provide 92% accuracy.

The problem of tree counting was resolved by preparing a small dataset of aerial images containing two classes with one class of image represents a tree in a picture while another class represents the presence of non-tree objects in the picture. On this dataset, a tree detection model was trained and according to the performance of the model, more samples of images were added to the dataset, which leads to increase in the performance of the model. Once the model is trained, it was used to count the trees from the image and achieved an R^2 of 0.95.

Thus, a conclusion can be made from Results that deep learning method gives more accurate tree count than morphological and color space methods. This study can also be helpful in multiple real-world Earth observation applications. It can also serve as baseline results for the future works. Complex models and training data can provide more accurate results.

REFERENCES

- Shi-Gang, Cui, Li Heng, Wu Xing-Li, Zhang Yong-Li, and He Lin. "Study on segmentation of lettuce image based on morphological reorganization and watershed algorithm." In *2018 Chinese Control and Decision Conference (CCDC)*. IEEE, 2018.
- Eddins, Steve. "The watershed transform: strategies for image segmentation." *The Mathworks Journal, Matlab®, News and Notes* (2002): 47-48.
- Hanbury, Allan. "Constructing cylindrical coordinate colour spaces." *Pattern Recognition Letters* 29, no. 4 (2008): 494-500.
- Joblove, George H., and Donald Greenberg. "Color spaces for computer graphics." In *ACM siggraph computer graphics*, vol. 12, no. 3, pp. 20-25. ACM, 1978.
- Mathworks.com. "Deep learning with matlab" 20 July 2018. <https://in.mathworks.com/solutions/deep-learning/convolutional-neural-network.html>
- Krizhevsky, Alex, Ilya Sutskever, and Geoffrey. "Imagenet classification with deep convolutional neural networks." In *Advances in neural information processing systems*, pp. 1097-1105. 2012.
- Crowther, T. W., H. B. Glick, K. R. Covey, C. Bettigole, D. S. Maynard, S. M. Thomas, J. R. Smith et al. "Corrigendum: Mapping tree density at a global scale." *Nature* 532, no. 7598 (2016): 268.
- Terol-Villalobos, Iván Ramon, and Damián Vargas-Vázquez. "Openings and closings with reconstruction criteria: a study of a class of lower and upper levelings." *Journal of Electronic imaging* 14, no. 1 (2005): 013.