Advanced Identification and Analysis of Forest Canopies in Satellite Imagery Using Deep Learning Algorithms

Harjinder Singh[1], Dr. Ranjana Sharma[2], Dr. Anil Kumar[3]

[1]Research Scholar, College of Computing Sciences & IT, Teerthanker Mahaveer University, Moradabad, India, Email: harjinder.mca07@gmail.com
[2]Associate Professor,College of Computing Sciences & IT, Teerthanker Mahaveer University, Moradabad, India, Email: sharmaranjana04@gmail.com
[3] Scientist/Engineer 'SF', Indian Institute of Remote Sensing, Dehradun, India, Email: aniliirsisro@gmail.com

Received: 22.05.2024 Revised: 15.07.2024 Accepted: 16.08.2024

ABSTRACT
Satellite imagery holds significance across various sectors such as disaster management, law enforcement, and environmental surveillance. These fields necessitate the manual recognition of elements and structures within the images. However, due to vast geographical areas and limited human resources, automation becomes essential. Traditional techniques to perform object detection and classification lack accuracy and dependability for this purpose. Deep learning, a wing of Artificial Intelligence emerged as a promising solution. It has demonstrated efficacy in understanding images through Convolutional neural network[1].Deep Convolutional Neural Networks (DCNNs) have shown outstanding results in most computer vision applications. Traditionally, a fixed data-set is used for detecting the object and other tasks, and after training, model is used exactly as is[2]. In this paper, the data-set consists of pictures of trees and other landscape-related categories that underwent the State-of-the-art CNN algorithm. We extracted a collection of tiles from satellite photos to provide a comprehensive view. And further, we have trained a CNN-based model and have successfully detected the bunch of trees with an accuracy of 95.62 % after fine-tuning. Further, we have trained AlexNet and LeNet models on the same data-set and compared the performance of the proposed model with these two benchmark deep learning models.

Keywords: Deep learning, Object Detection, CNN, Trees Detection, Artificial Intelligence, Remote Sensing

1. INTRODUCTION

Trees are the garland of our earth. Starting from food, wood even CO2 is special contribution of trees to the mankind. Their presence is very crucial and important for the human survival. Trees nourish us with fresh air to breathe, pure water to drink, shade, and food for us, the animals, and the plants. In addition Trees have recreational, cultural and spiritual significance mentioned in Table 1.

Table 1 proves that Trees are critical part of mankind therefore; Trees require our constant care and preservation since they are vital to the health of the planet's ecosystem and the animals which consider it home. Having accurate and up-to-date information about the geographical distribution of vulnerable tree species in the area is crucial for their survival. Remote sensing techniques are increasingly replacing expensive and time-consuming field surveys for forest resource evaluation. For this reason, satellite, aerial, and, UAV (unmanned aerial vehicles) have been very popular and effectively used platforms for data collection[3] .

If we are really interested in any object of our interest, and then capturing images is a good choice in order to perform any experiment related to computer vision. Our Interest of object is trees. Objective is to train a model which can detect tree automatically.

Table 1: Contribution of Trees in different aspects of Life.

Aspects of Life	Contribution of Trees to humanity
Social and Community Values	Vintage trees that are a wonderful source of community pride and historic markers. Trees surround backyards, parks, streets, and playgrounds, creating a visually pleasing and tranquil environment.

Environmental and Ecological Value	In addition to supplying oxygen and improving air quality and reducing the effects of climate change, trees also protect soil, save water, and help wildlife. An acre of forest emits four tons of oxygen and absorbs six tons of carbon dioxide, according to the U.S. Department of Agriculture [4].
Personal & Spiritual Value	The primary reason we adore trees is their natural beauty and grandeur A apparently infinite range of shapes, patterns, textures, and vivid colors are displayed
Commercial & Personal Value	Timber from trees is used to make hundreds of household products, tools, recreational goods, furniture, and buildings. Paper is made from wood pulp.
Economical Value	Trees and bushes by itself are valuable and can save money.

There are mainly 3 ways to capture the image of objects in real world shown in Figure 1. For this research we have taken the data-set of images taken from satellite using remote sensing.

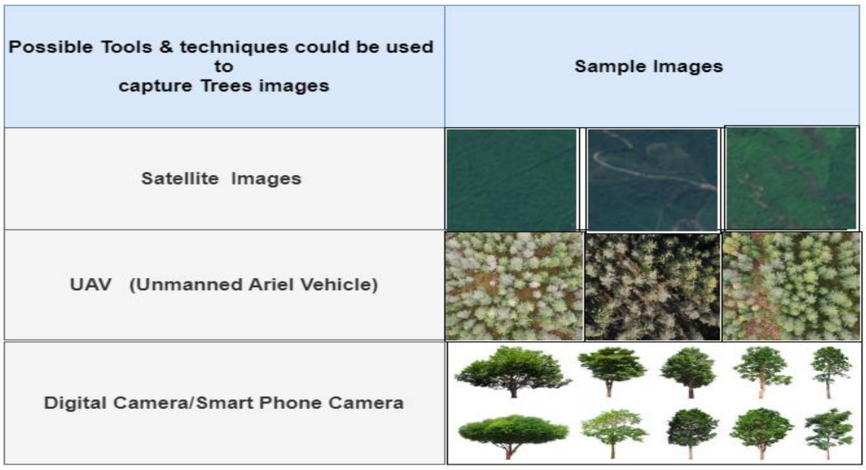

Figure 1: Images of trees captured through different tools & techniques

Remote sensing technique (capturing images from a far distance without being physically in touch with the help of sensors/satellites) has come up as a hot spot for computer vision research, is to identify and categorize objects that are relevant in remote sensing images[5–8]. Deep learning method is used to detect and predict the object in remote sensing shots. With numerous of possible uses, such as autonomous driving , face recognition , pedestrian identification , medical diagnosis , and more, remote sensing image object detection has proven the capability to perform extremely challenging tasks[5].

Since object detection in aerial photography is an interesting problem, the fields of computer vision and machine learning based research have been attracted towards this area. These techniques usually employ Convolutional Neural Networks (CNN) for object detection. The current focus has been on identifying automobiles, highways, agriculture, health care, and other things [7–15]. The proposed work stats that we wish to develop and use a framework to detect the existence of trees by implementing the classical algorithm CNN. The suggested plan of action for the detection of trees offers a starting point that is readily expandable to include the detection and identification of individual tree detection and different tree species based classification.

The remaining portion of this paper is framed as follows: We give a brief overview of deep learning methods for objection identification in Section II. We go into great depth on the training process, techniques, and data-set utilized in this study in Section III. With the aid of figures, we provide the results of our tree detection accuracy and scope of improvement in Section IV. We also offer a comparative analysis and discussion related the proposed model, popular CNN models and their performance outcomes ,future scope in Section V,VI and VII sections.

2. Related Work

Deep learning become so popular recently because of the explosion of data brought forth by IOT devices, social media, and IT usage, deep learning algorithms have the ability to generate an enormous amount of actual data [16]. Where, the size of the training data are large, the true potential of neural networks and deep learning is shown. Deep learning training may be completed in a respectable period of time thanks to improvements in hardware, such as GPU and TPU's, which enable one to do a great number of calculations in parallel. On the other hand, Python and the open-source ecosystem have made it easier for anyone without programming experience to develop deep learning applications since they can use Python with PyTorch or Tensor flow. It is not compulsion now days to purchase costly hardware to develop machine learning programs; one may rent a computer in the cloud. These days, artificial intelligence is seeing a growth in enterprises, and many CEO's want to take advantage of this. This quickens the development of deep learning even further.

In the last few decades, satellite imaging has been employed for a wide variety of purposes, including target identification, regional planning, forestry, agriculture. In order to study the effects of natural emergencies and other unfavorable events on the environment, satellite photography has also been widely used[8].

Lucas Santos Santana et al.(2023) implemented with the help of Open Source Computer Vision Library and a CNN architecture served worked to be a foundation of this approach. Three, six, and twelve months following the initial planting, the assessments were conducted in coffee-growing regions at various phases of development. The sorted data set has been used to train “You Only Look Once (YOLOv3)” neural network after photos were obtained. Utilizing 7458 plants that were three, six, and twelve months old, the training stage was carried out, stabilizing in iterations between 3000 and 4000.Due to crown unification, it was impossible to recognize the plant within a year. With plants that were three months old, a counting accuracy of 86.5% was attained. The decrease in accuracy may have been influenced by the properties of the plants at this stage, and the low canopy uniformity may have made it difficult for the neural network to identify a pattern. For automatic plant counting in plantations with six months of development, 96.8% accuracy was attained[15].

Nurhabib et al. (2022) has presented a model to analyze the photographs taken by a satellite from Google Earth, a YOLO algorithm has been used in this study to explore the method of detecting and counting oil palm plants. The system prototype was successfully constructed and tested to identify and count oil palm trees in Oil Palm Teaching Farm. Using YOLO with 2500 step iterations and a loss value of 0.6, the training dataset produced accuracy scores of 85.6%, 98.9% for precision, and 86.6% for recall based on the experimental findings. This demonstrates that the model is capable of accurately identifying and counting palm trees. Future improvements are also suggested, including the use of ‘image-correction’ techniques with selection of different satellites to provide better photos [12].

Abdur Rahman et al. (2022) illustrated in this study that 848 color photos of weeds were gathered in fields of cotton under various conditions of field and organized a dataset which contains three classes of weed and annotations files which represents their bounded box parameters. By applying the object identification models that had been pre trained with the discussed dataset,RetinaNet,YOLOv5, Fast RCNN, EfficientDet , Faster RCNN were created. With 79.98% of (mAP), RetinaNet (R101-FPN), although having a long inference time, had the maximum overall accuracy of detection [17].

In this work, Eduardo T. Assunção et al. (2022) proposed a deep learning model build using faster RCNN algorithm to perform fruit detection on the peach trees for the yield estimation to promote precision agriculture. The research started with images collection for the preparation of dataset followed by an object detection framework (faster RCNN) to build the model. After Training and testing, Evaluation metric IOU has been defined to evaluate the task performed. Model performed well with the accuracy of 90%. In future work, In order to find hidden fruits, one can include photographs from various orchards and move the images nearer to the tree to prepare dataset which can improve the accuracy of the model in case of hidden fruits behind leaf cover[13].

Boon Ho et al. (2022) shown how two novel online frameworks were used by this study to analyze and rank the impact of each single tree canopy cover loss. Individual trees are localized and categorized into the appropriate crown loss percentage bins using the real-time crown loss estimating algorithm. As genuine photos from various perspectives are typically expensive to gather, experiments are being done to see if artificially generated images of trees could be utilized for training of the RTCLE model. Results indicate that training with synthetic data is useful to maintain a threshold (mAP) mean average precision that may be increased in future with only a little amount of real picture data. By combining the genuine data-set with the created synthetic data, we demonstrated that the mAP may be raised by roughly 60% to 78% [9].

After reviewing the literature we came on this conclusion that most of the researchers have applied CNN based algorithms for object detection and classification. If someone wants to focus on binary classification then CNN Family algorithms could be applied and if Multi class classification is required then YOLO Family algorithms are far most suited algorithms. To evaluate the accuracy of the model, confusion matrix could be draw and on the basis of that one can calculate various scores in percentage like precision, recall and F1 score etc.
In our research, as we have focused on binary classification, State of art CNN algorithm has been used to achieve the classification results with accuracy at higher side.

3. MATERIALS AND METHODS

Figure 2 gives a synopsis of the entire process. To put it briefly, we divide the data into training and test sets after it has been prepared. Since no per-trained weights are used, it is necessary to the overall training of any neural network. Lastly, we evaluate the overall performance using test data which was not accessible before. We discuss each of these stages in more depth in the following subsections.

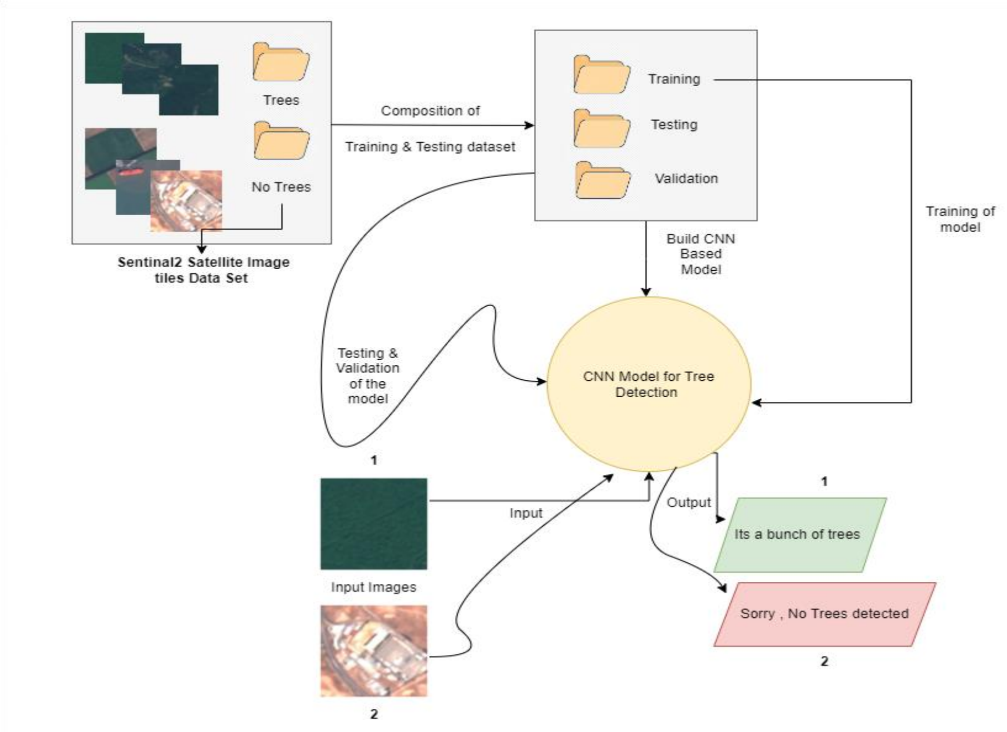

Figure 2: Research Methodology streamlined from left to right.

Data Collection and Pre-processing: To build any deep learning based model, there is a requirement of numerous of images for the training purpose. To do the same we have taken 10,400 images in the form of image chips of size 64X64 which has been cropped from satellite images which are taken from sentinel 2 satellites. Figure 3.1, 3.2, 3.3, 3.4 reveal few of the images from the discussed data-set has been used for this research.

Figure 3.1: Images of Forest Region

Figure 3.2: Images of Agricultural Land

Figure 3.3: Images of Industrial Area

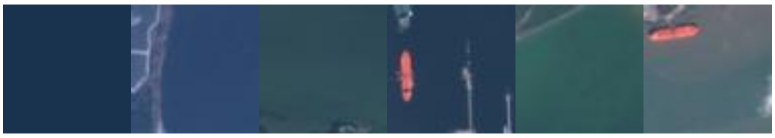

Figure 3.4: Images of Sea Area

Data-set Source:www.kaggle.com
Satellite: https://sentinel.esa.int/web/sentinel/missions/sentinel-2
In the data-set we have two classes. First is Trees under which we have 5200 images of Forest region and another is No Trees under which we have 5200 images of Agricultural, Industrial and Sea Area as show-cased in Figure 4.

Figure 4: Classes of Data-set

This data-set was further pre-processed into the directory structure shown in Figure5, so that model could be trained.

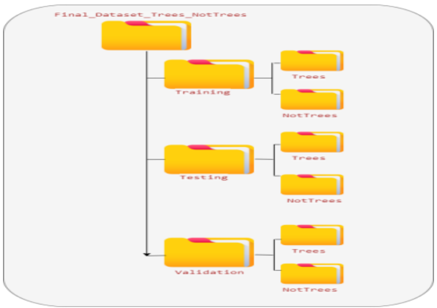

Figure 5: Data-set directory structure

We have divided the data-set into three parts and i.e. Training, Testing and Validation folder. 80% of the images are going to be used for training of the model and images are stored in Training folder as per figure 5. 10% of the images are going to be used for testing of the model and images are stored in Testing folder as per figure 5. 10% of the images are going to be used for validation of the model and images are stored in Validation folder as per figure 5. Above stated folders individually has two different folders as per classes defined in figure 4.
To build numpy arrays for training, testing and validation as mentioned in below snippet taken from our Google Colab notebook was furthered processed from the above directory structure.

```
train = create_training_data('/content/drive/MyDrive/Final_Dataset_Trees_NotTrees/Training')
test = create_training_data('/content/drive/MyDrive/Final_Dataset_Trees_NotTrees/Testing')
val = create_training_data('/content/drive/MyDrive/Final_Dataset_Trees_NotTrees/Validation')
```

Building Deep Learning model and Training: Deep learning model can be visualized as an artificial neural network with multiple layers that can recognize hierarchical data representations. These models have proved to be successful in various tasks like natural language processing, audio and picture recognition, and more. The models' depth allows them to automatically gather information at various levels of abstraction, which helps them identify complex patterns within the input data [18].
There is one of the types of deep learning model that works especially well for processing grid-like input such as photographs. This model is called the Convolutional neural network (CNN). To understand CNN,

one needs to have knowledge of calculus, optimization, and linear algebra. Here is a brief explanation of the fundamental mathematical concepts used in CNN.

Let's assume that I represent the input feature map (image) of a layer. We can represent the Convolutional kernel or filter of the input as K. The convolution process can be denoted by the symbol *, and convolution layer's output can be framed as $S = I * K$. To carry out the convolution process, we slide the kernel across the input and calculate the element-wise product and sum for each position[19].

To introduce non-linearity to the model, an activation function is implemented element-by-element after the convolution process. Few activation functions list include Tanh, sigmoid, and ReLU (Rectified Linear Unit). Let activation function be denoted by $\sigma(x)$. The active output can be calculated through the formula $A=\sigma(S)$.

There is a technique called pooling through which we can reduce the spatial dimensions of an input. The most common method is max pooling, which involves selecting the largest value in a particular local area. Another method is average pooling, which computes the average value in a specific region. Let's assume that P represents the pooling function. The resulting output after pooling is denoted as $P(A)$.

When combined, the convolution, activation, and pooling processes result in the following mathematical representation for a single layer of a CNN: $P(\sigma(I*K))$ Where P represents Pooling , σ represents Activation function, I is the input, and K is the filter [20].

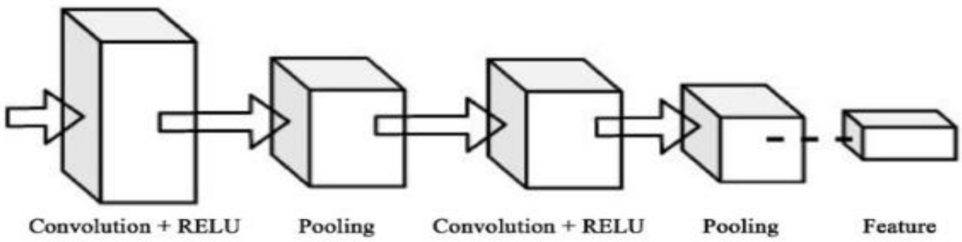

Figure 6: Basic Block diagram of CNN architecture [21–33]

With the help of above described algorithm and CNN basic architecture given in figure 6, CNN based classification model has been developed. The summary of CNN Based Classification model to detect Trees and Not Trees along with its architecture depicted in Figure 7 and Table 2.

Table 2: Summary of proposed CNN Based Classification model

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 64, 64, 1)]	0
conv2d (Conv2D)	(None, 64, 64, 32)	320
batch_normalization (Batch Normalization)	(None, 64, 64, 32)	128
conv2d_1 (Conv2D)	(None, 64, 64, 32)	9248
batch_normalization_1 (Batch Normalization)	(None, 64, 64, 32)	128
max_pooling2d (MaxPooling2D)	(None, 32, 32, 32)	0
conv2d_2 (Conv2D)	(None, 32, 32, 64)	18496
batch_normalization_2 (Batch Normalization)	(None, 32, 32, 64)	256
conv2d_3 (Conv2D)	(None, 32, 32, 64)	36928
batch_normalization_3 (Batch Normalization)	(None, 32, 32, 64)	256
max_pooling2d_1 (MaxPooling2D)	(None, 16, 16, 64)	0
conv2d_4 (Conv2D)	(None, 16, 16, 128)	73856
batch_normalization_4 (Batch Normalization)	(None, 16, 16, 128)	512
conv2d_5 (Conv2D)	(None, 16, 16, 128)	147584
batch_normalization_5 (Batch Normalization)	(None, 16, 16, 128)	512
max_pooling2d_2 (MaxPooling2D)	(None, 8, 8, 128)	0
conv2d_6 (Conv2D)	(None, 8, 8, 256)	295168
batch_normalization_6 (Batch Normalization)	(None, 8, 8, 256)	1024
conv2d_7 (Conv2D)	(None, 8, 8, 256)	590080
batch_normalization_7 (Batch Normalization)	(None, 8, 8, 256)	1024
max_pooling2d_3 (MaxPooling2D)	(None, 4, 4, 256)	0
flatten (Flatten)	(None, 4096)	0
dense (Dense)	(None, 512)	2097664
dropout (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 512)	262656
dropout_1 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 512)	262656
dropout_2 (Dropout)	(None, 512)	0
dense_3 (Dense)	(None, 1)	513

Total params: 3,799,009

Trainable params: 3,797,089

Non-trainable params: 1,920

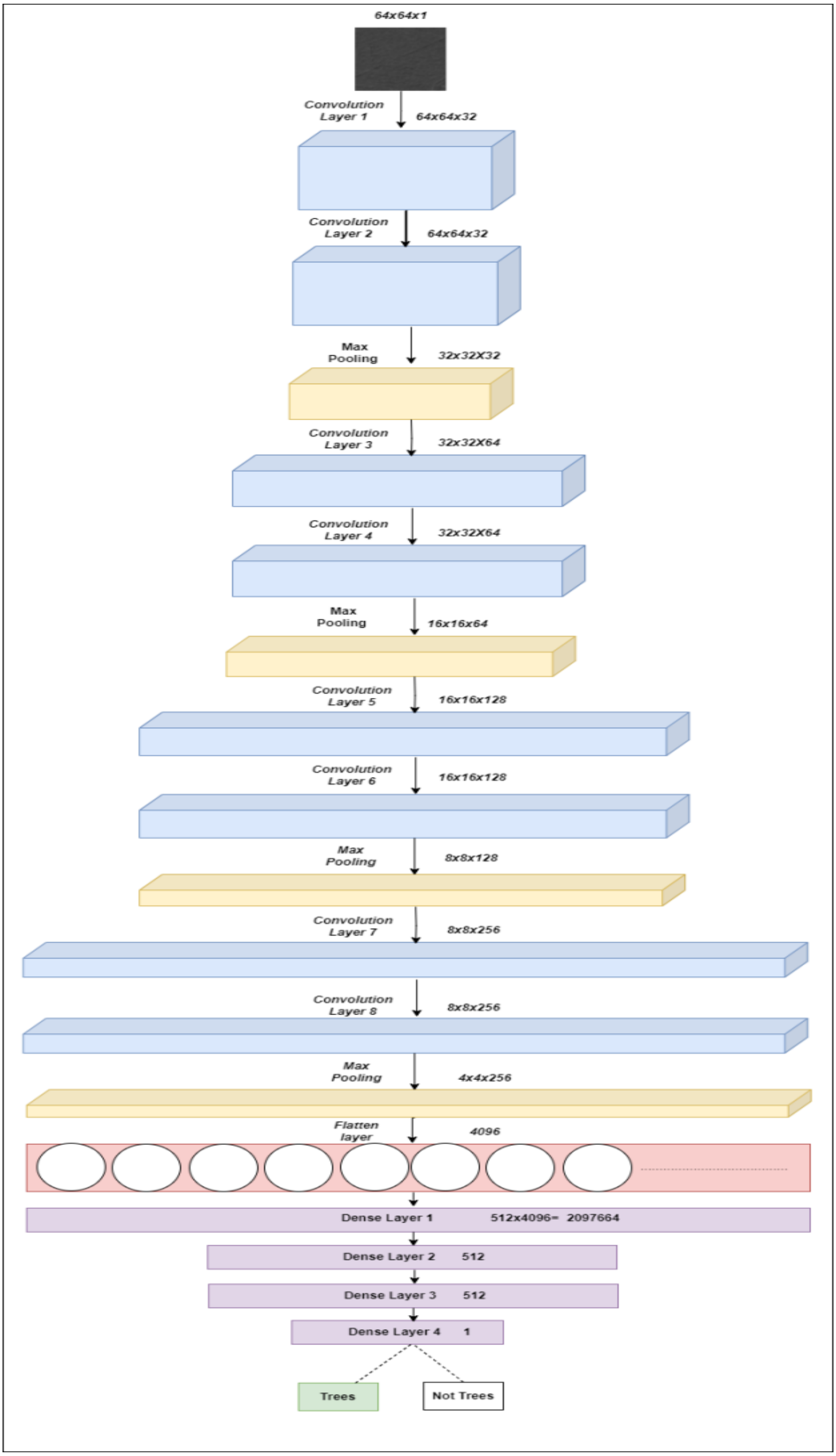

Figure 7: Proposed CNN Model

For every layer mentioned in Table 2 & Figure 7, for the input size for each and every layer except the input image is calculated through the Following formulas.
Formula to calculate numbers of parameters calculation for Convolutional layer:

$$Paramsconv=Cin \times Cout \times Kheight \times Kwidth+Cout$$

Where Paramsconv = formula for parameter calculation for Convolutional layer, Cin= number of channels of the input image,Cout= number of channels in the output image, Kheight= height of the kernel,Kwidth= kernel width.
Formula to calculate numbers of parameters calculation for fully connected layer:

$$Paramsfc=Nin \times Nout+Nout$$

Where Paramsfc= formula for parameter calculation for fully connected layer, Nin= Input neurons in fully connected layer Nout= Output neurons in fully connected layer.
Implementation Setup: To develop the suggested model, Keras and Tensorflow with Python 3 in Jupyter Notebook were used. For Training of the model and result analysis, we have used Google Colab.Table 3 displays the parameters that are utilized to effectively learn the model.

Table 3: Important Parameters considered while training of the model

S.No.	Parameters	Values
1	CNN Layers	8
2	Max Pooling Layers	4
4	Epochs	20
5	Batch Size	6
6	Dropout Rate	0.1
7	Optimizer	Adam
8	Loss Function	Binary Cross Entropy

Once your model is developed and trained then to check its performance, we have to apply some evaluation test .Table 4 shows common criteria for assessing the performance of a CNN classification model. If these scores are good then Model is feasible.

Table 4: Performance assessment techniques applied on the trained model.

Performance Evaluation Tests	Formulas
Precision	TP/(TP+FP)
Recall	TP/(TP+FN)
Accuracy	(TP+TN)/(TP+TN+FP+FN)
F1- Score	2(Precision X Recall)/(Precision – Recall)
Macro-F1	Average of F1 Scores
Weighted-F1	Weighted-averaged of F1 Scores
Miss-classification Rate	Incorrect predictions/ Total predictions
TP= True Positive , TN= True Negative FP= False Positive, FN= False Negative	

4. RESULTS

The data-set used in this research work, comprising satellite images, used for training of model which can further classify two categories: Trees and Not Trees. This section presents our findings.
We used our trained model's weights to detect trees after 5 epochs of training in consecutive 6 batches. Screenshot of one of the final batch under training is depicted below.

```
Epoch 1/5
1873/1873 [==============================] - ETA: 0s - loss: 0.2998 - accuracy: 0.8981
Epoch 1: val_loss improved from inf to 0.24283, saving model to harry_trees02.h5
1873/1873 [==============================] - 592s 314ms/step - loss: 0.2998 - accuracy: 0.8981 - val_loss: 0.2428 - val_accuracy: 0.9248
Epoch 2/5
1873/1873 [==============================] - ETA: 0s - loss: 0.2564 - accuracy: 0.9135
Epoch 2: val_loss improved from 0.24283 to 0.19116, saving model to harry_trees02.h5
1873/1873 [==============================] - 567s 302ms/step - loss: 0.2564 - accuracy: 0.9135 - val_loss: 0.1912 - val_accuracy: 0.8997
Epoch 3/5
1873/1873 [==============================] - ETA: 0s - loss: 0.2276 - accuracy: 0.9202
Epoch 3: val_loss did not improve from 0.19116
1873/1873 [==============================] - 552s 295ms/step - loss: 0.2276 - accuracy: 0.9202 - val_loss: 0.2891 - val_accuracy: 0.8597
Epoch 4/5
1873/1873 [==============================] - ETA: 0s - loss: 0.2080 - accuracy: 0.9299
Epoch 4: val_loss did not improve from 0.19116
1873/1873 [==============================] - 551s 294ms/step - loss: 0.2080 - accuracy: 0.9299 - val_loss: 0.2802 - val_accuracy: 0.9007
Epoch 5/5
1873/1873 [==============================] - ETA: 0s - loss: 0.1853 - accuracy: 0.9331
Epoch 5: val_loss improved from 0.19116 to 0.11504, saving model to harry_trees02.h5
1873/1873 [==============================] - 548s 292ms/step - loss: 0.1853 - accuracy: 0.9331 - val_loss: 0.1150 - val_accuracy: 0.9562
```

After completion of training of the model, we came up with confusion matrix depicted in figure 8 which clearly indicates that model is performing pretty well. Successful predictions numbers are higher either True Positive or True Negative.

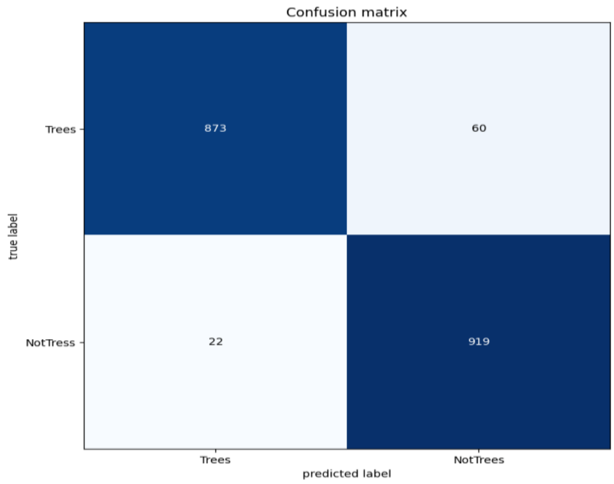

Figure 8: Confusion Matrix of the proposed CNN Model

Accuracy graphs were plotted after seeing the satisfying results in confusion matrix. The graphs clearly presented the facts in Figure 9, 10 that when Training is going on of the CNN model as the epochs are getting incremented, the loss decreased and accuracy is increased. These graphs have been plotted with no of epochs on X axis and Accuracy/Loss percentage on Y axis.

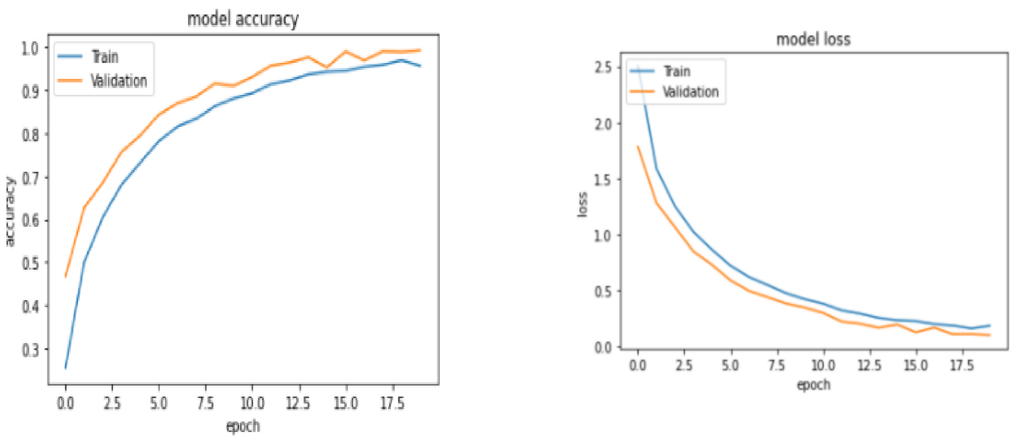

Figure 9, 10: Training& Validation Loss graph, Training & Validation Accuracy graph

The proposed model came up with 95.62% of accuracy and F1 score 95.51% for classifying Trees and 95.73% for classifying Not Trees category of images. Detailed Matrix is given in Figure 11.

Class Name	Precision	1-Precision	Recall	1-Recall	f1-score
Trees	0.9357	0.0643	0.9754	0.0246	0.9551
Not Trees	0.9766	0.0234	0.9387	0.0613	0.9573
Accuracy	0.9562				
Misclassification Rate	0.0438				
Macro-F1	0.9562				
Weighted-F1	0.9563				

Figure 11: Performance Evaluation matrix of Proposed CNN Classification Model

Table 5 displays a comparative analysis of different CNN architectures, such as Alex Net, Le Net.

Table 5: Performance Comparison table

Accuracy Metric	Alex Net	Le Net	Proposed Model
Overall Accuracy	78.98%	89.16%	95.62%
F1 Score	81.04%	88.60%	95.51%

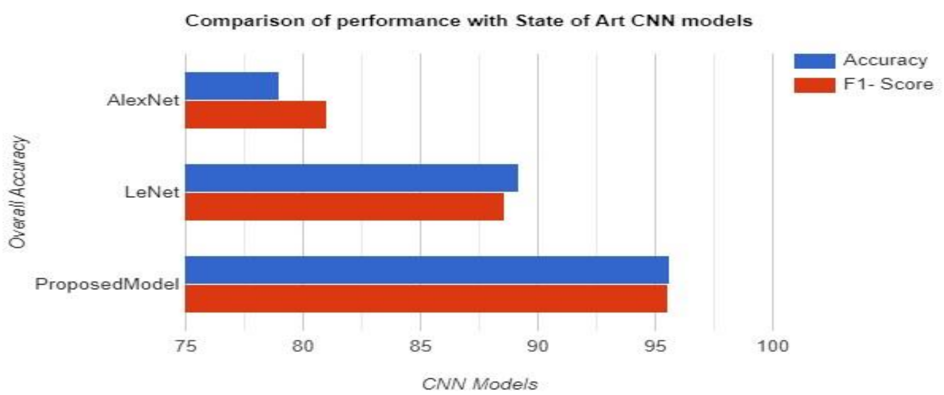

Figure 12: Comparison of Performance of Proposed model with other few Benchmark CNN models

Figure 12 shows that the proposed CNN classifier's overall accuracy is 95.62%, with a 95.51% F1Score. The following Table 5 compares various CNN architectures, including Le Net and Alex Net, displaying their overall accuracy as well as their F1 score.
In short, using tiles from a collection of remotely sensed aerial photography, the research project is an attempt to categorize and identify a group of trees. Further research in this field has a great deal of promise. Getting a cleaner data-set and developing techniques for better annotations are important tasks since they will enhance model training.

5. DISCUSSION

In this research project, we employed a total of 1040 testing samples of satellites images tiles that were randomly chosen from a larger set of 10400 overall samples. Our primary objective was to evaluate the classification accuracy of our CNN model. We found that various factors, including max-pooling and Convolutional kernel sizes, as well as the number of fully connected layers, have a significant impact on the accuracy of the model. Model comprises of 8 Convolutional layers along with 4 max pooling layers.In order to enhance neural network training in deep learning, batch normalization is employed. Its primary goals are to: reduce the need for meticulous initialization; stabilize the learning process; provide greater learning rates; and serve as a regularizer. In this research paper, Table 1 shows the contribution of trees in different aspects of life which motivated us to choose trees as an object for detection so that surveillance and monitoring of the trees could be automated. Table 2 contains the summary of the proposed CNN model. Table 3

Represents the important parameters considered while developing and training of the model. Once model got trained, to assess the performance of the model, various techniques were used which are mentioned in Table 4. In last Table 5 shows the performance comparison of the proposed model with AlexNet and LeNet models. Figure 1 depicts the different tools and techniques to capture images followed by Figure 2 reflect the research methodology applied to propose my AI based model. Figure 3.1, 3.2, 3.3, 3.4 shows glimpses of dataset used for training and testing of the model. Figure 4 and 5 represents the classes of the dataset and directory structure setup of dataset. Figure 6 shows the generic CNN Model architecture. Figure 7 represents the Layered architecture of proposed model. After training of the model, we validated the results using testing and validation images and came up with confusion matrix which has been shown in Figure 8.Figure 9,10 graphs shows the model accuracy and model loss. Figure 11 shows the accuracy percentage of different performance evaluation formulas like precision, F1 score etc.

The experimental results indicate that we achieved a maximum overall accuracy of 95.62% after fine tuning of the model. After that we analyzed the results through drawing confusion matrix and applied various evaluation formulas to check the accuracy of the model such as F1 score, Precision and recall. Model performed well and produced F1 Score as 95.73% for predicting testing image as Not Trees category and produced 95.51% for predicting testing image as Trees category. Other scores are mentioned in Figure 12.

After that we did comparative study of our CNN Model with two renowned models that is AlexNet and LeNet. For the same dataset, AlexNet‘s accuracy score was 78.98% and LeNet‘s accuracy score was 89.16%

6. CONCLUSION

In this research project, 3 CNN-based object detection models were trained and tested using tiles of satellite images taken by the Sentinel2 satellite. The Alex net and Le Net model under performed for the data-set used in the study. However, the proposed CNN model which used grayscale images as input, performed well in the experiments. This model produced the most accurate results with an average precision of 95.62%. It is also one of the observations that with the same architecture, if we vary the number of neurons in the dense layer, there is no significant difference in the training time. Although that is true that proposed model has a lot of trainable parameters, but it was faster than other 2 models. We have achieved the highest accuracy in detection of bunch of trees with the use of the proposed model. There are few limitations of this research is that there are many hidden layers we have used to fine tune the model. High performance GPU’s are required to make model trained. Prediction accuracy & time taken for prediction is nice but Training time is at higher side.

7. Future Scope

For the future projection, we wish to work towards object detection of individual tree detection and further classify them on the basis of tree species. For that we need to work the data-set of those images which are snapped from lesser altitude & high resolution.

Author Contributions:Conceptualization, H.S.,R.S. and A.K.; Methodology, H.S.,R.S. andA.K.; software, H.S.,R.S.,A.K.; validation, H.S.,R.S.,A.K.; formal analysis, H.S..; investigation, H.S.,R.S. and A.K.; resources, H.S.,R.S. and A.K.; data Curation, H.S.,R.S. and A.K.; writing—original draft preparation, H.S.; writing—review and editing, H.S.,R.S. and A.K.; visualization, H.S.,R.S. and A.K.; supervision, R.S. and A.K.; project administration, H.S.,R.S.,A.K.; All authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] Mohanty, S.P.; Czakon, J.; Kaczmarek, K.A.; Pyskir, A.; Tarasiewicz, P.; Kunwar, S.; Rohrbach, J.; Luo, D.; Prasad, M.; Fleer, S.; et al. Deep Learning for Understanding Satellite Imagery: An Experimental Survey. Front. Artif. Intell.2020, 3, 534696.

[2] Zhao, H.; Morgenroth, J.; Pearse, G.; Schindler, J. A Systematic Review of Individual Tree Crown Detection and Delineation with Convolutional Neural Networks (CNN). Curr. For. Rep.2023, 9, 149–170.

[3] Al-Mulla, Y.; Ali, A.; Parimi, K. Deep Learning Techniques on Very High Resolution Images for Detecting Trees and Their Health Conditions. In Proceedings of the IGARSS 2023 - 2023 IEEE International Geoscience and Remote Sensing Symposium; 2023; pp. 6542–6544.

[4] Tao, H.; Li, C.; Zhao, D.; Deng, S.; Hu, H.; Xu, X.; Jing, W. Deep Learning-Based Dead Pine Tree Detection from Unmanned Aerial Vehicle Images. Int. J. Remote Sens.2020, 41, 8238–8255.

[5] Miyoshi, G.T.; Arruda, M. dos S.; Osco, L.P.; Marcato Junior, J.; Gonçalves, D.N.; Imai, N.N.; Tommaselli, A.M.G.; Honkavaara, E.; Gonçalves, W.N. A Novel Deep Learning Method to Identify Single Tree Species in UAV-Based Hyperspectral Images. Remote Sens.2020, 12, 1294.

[6] Xiong, Y.; Zeng, X.; Lai, W.; Liao, J.; Chen, Y.; Zhu, M.; Huang, K. Detecting and Mapping Individual Fruit Trees in Complex Natural Environments via UAV Remote Sensing and Optimized YOLOv5. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.2024, 17, 7554–7576, doi:10.1109/JSTARS.2024.3379522.

[7] Zhang, L.; Lin, H.; Wang, F. Individual Tree Detection Based on High-Resolution RGB Images for Urban Forestry Applications. IEEE Access2022, 10, 46589–46598, doi:10.1109/ACCESS.2022.3171585.

[8] Kwenda, C.; Gwetu, M.; Dombeu, J.V.F. Machine Learning Methods for Forest Image Analysis and Classification: A Survey of the State of the Art. IEEE Access2022, 10, 45290–45316, doi:10.1109/ACCESS.2022.3170049.

[9] Ho, B.; Kocer, B.B.; Kovac, M. Vision Based Crown Loss Estimation for Individual Trees with Remote Aerial Robots. ISPRS J. Photogramm. Remote Sens.2022, 188, 75–88.

[10] Nezami, S.; Khoramshahi, E.; Nevalainen, O.; Pölönen, I.; Honkavaara, E. Tree Species Classification of Drone Hyperspectral and RGB Imagery with Deep Learning Convolutional Neural Networks. Remote Sens.2020, 12, 1070.

[11] Gan, Y.; Wang, Q.; Iio, A. Tree Crown Detection and Delineation in a Temperate Deciduous Forest from UAV RGB Imagery Using Deep Learning Approaches: Effects of Spatial Resolution and Species Characteristics. Remote Sens.2023, 15, 778.

[12] Nurhabib, I.; Seminar, K.; others Recognition and Counting of Oil Palm Tree with Deep Learning Using Satellite Image. In Proceedings of the IOP Conference Series: Earth and Environmental Science; IOP Publishing, 2022; Vol. 974, p. 012058.

[13] Assunção, E.T.; Gaspar, P.D.; Mesquita, R.J.; Simões, M.P.; Ramos, A.; Proença, H.; Inacio, P.R. Peaches Detection Using a Deep Learning Technique—A Contribution to Yield Estimation, Resources Management, and Circular Economy. Climate2022, 10, 11.

[14] Yarak, K.; Witayangkurn, A.; Kritiyutanont, K.; Arunplod, C.; Shibasaki, R. Oil Palm Tree Detection and Health Classification on High-Resolution Imagery Using Deep Learning. Agriculture2021, 11, 183.

[15] Santana, L.S.; Ferraz, G.A. e S.; Santos, G.H.R. dos; Bento, N.L.; Faria, R. de O. Identification and Counting of Coffee Trees Based on Convolutional Neural Network Applied to RGB Images Obtained by RPA. Sustainability2023, 15, 820.

[16] Wang, H.; Li, J.; Van de Voorde, T.; Zhou, C.; De Maeyer, P.; Ma, Y.; Shen, Z. Individual Populus Euphratica Tree Detection in Sparse Desert Forests Based on Constrained 2-D Bin Packing. IEEE Trans. Geosci. Remote Sens.2024, 62, 1–19, doi:10.1109/TGRS.2024.3391352.

[17] Rahman, A.; Lu, Y.; Wang, H. Performance Evaluation of Deep Learning Object Detectors for Weed Detection for Cotton. Smart Agric. Technol.2023, 3, 100126.

[18] Lumnitz, S.; Devisscher, T.; Mayaud, J.R.; Radic, V.; Coops, N.C.; Griess, V.C. Mapping Trees along Urban Street Networks with Deep Learning and Street-Level Imagery. ISPRS J. Photogramm. Remote Sens.2021, 175, 144–157.

[19] Prabowo, Y.; Pradono, K.A.; Amriyah, Q.; Rasyidy, F.H.; Carolita, I.; Setiyoko, A.; Candra, D.S.; Musyarofah; Ulfa, K.; Hestrio, Y.F. Palm Trees Counting Using MobileNet Convolutional Neural Network in Very High-Resolution Satellite Images. In Proceedings of the 2022 IEEE Asia-Pacific Conference on Geoscience, Electronics and Remote Sensing Technology (AGERS); 2022; pp. 79–83.

[20] Hao, Z.; Lin, L.; Post, C.J.; Mikhailova, E.A.; Li, M.; Chen, Y.; Yu, K.; Liu, J. Automated Tree-Crown and Height Detection in a Young Forest Plantation Using Mask Region-Based Convolutional Neural Network (Mask R-CNN). ISPRS J. Photogramm. Remote Sens.2021, 178, 112–123.

[21] A Systematic Review of Individual Tree Crown Detection and Delineation with Convolutional Neural Networks (CNN) | Current Forestry Reports Available online: https://link.springer.com/article/10.1007/s40725-023-00184-3 (accessed on 19 February 2024).

[22] Agriculture | Free Full-Text | Oil Palm Tree Detection and Health Classification on High-Resolution Imagery Using Deep Learning Available online: https://www.mdpi.com/2077-0472/11/2/183 (accessed on 19 February 2024).

[23] ZamboniThgeThe, P.A.P.; Marcato, J.; Miyoshi, G.T.; de Andrade Silva, J.; Martins, J.; Gonçalves, W.N. Assessment of CNN-Based Methods for Single Tree Detection on High-Resolution RGB Images in Urban Areas. In Proceedings of the 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS; July 2021; pp. 590–593.
[24] Santos, A.A. dos; Marcato Junior, J.; Araújo, M.S.; Di Martini, D.R.; Tetila, E.C.; Siqueira, H.L.; Aoki, C.; Eltner, A.; Matsubara, E.T.; Pistori, H.; et al. Assessment of CNN-Based Methods for Individual Tree Detection on Images Captured by RGB Cameras Attached to UAVs. Sensors2019, 19, 3595, doi:10.3390/s19163595.
[25] Hao, Z.; Lin, L.; Post, C.J.; Mikhailova, E.A.; Li, M.; Chen, Y.; Yu, K.; Liu, J. Automated Tree-Crown and Height Detection in a Young Forest Plantation Using Mask Region-Based Convolutional Neural Network (Mask R-CNN). ISPRS J. Photogramm. Remote Sens.2021, 178, 112–123, doi:10.1016/j.isprsjprs.2021.06.003.
[26] Zortea, M.; Macedo, M.M.G.; Mattos, A.B.; Ruga, B.C.; Gemignani, B.H.; Paster, A. Automatic Citrus Tree Detection from UAV Images Based on Convolutional Neural Networks.
[27] Li, W.; Fu, H.; Yu, L. Deep Convolutional Neural Network Based Large-Scale Oil Palm Tree Detection for High-Resolution Remote Sensing Images. In Proceedings of the 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS); July 2017; pp. 846–849.
[28] Li, W.; Fu, H.; Yu, L.; Cracknell, A. Deep Learning Based Oil Palm Tree Detection and Counting for High-Resolution Remote Sensing Images. Remote Sens.2017, 9, 22, doi:10.3390/rs9010022.
[29] Jintasuttisak, T.; Edirisinghe, E.; Elbattay, A. Deep Neural Network Based Date Palm Tree Detection in Drone Imagery. Comput. Electron. Agric.2022, 192, 106560, doi:10.1016/j.compag.2021.106560.
[30] Zheng, J.; Li, W.; Xia, M.; Dong, R.; Fu, H.; Yuan, S. Large-Scale Oil Palm Tree Detection from High-Resolution Remote Sensing Images Using Faster-RCNN. In Proceedings of the IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium; July 2019; pp. 1422–1425.
[31] Recognition and Counting of Oil Palm Tree with Deep Learning Using Satellite Image - IOPscience Available online: https://iopscience.iop.org/article/10.1088/1755-1315/974/1/012058/meta (accessed on 19 February 2024).
[32] Symmetry | Free Full-Text | Identification of Apple Tree Leaf Diseases Based on Deep Learning Models Available online: https://www.mdpi.com/2073-8994/12/7/1065 (accessed on 19 February 2024).
[33] Remote Sensing | Free Full-Text | Large Scale Palm Tree Detection in High Resolution Satellite Images Using U-Net Available online: https://www.mdpi.com/2072-4292/11/3/312 (accessed on 19 February 2024).