Urban Tree Detection from Remote Sensing Data based on DeepForest Model

Vasil Dakov[a, *], Dessislava Petrova-Antonova [b]

[a]Delft University of Technology, Delft, The Netherlands - V.D.Dakov@student.tudelft.nl
[b]GATE Institute, Sofia University "St. Kliment Ohridski", Sofia, Bulgaria - dessislava.petrova@gate-ai.eu

KEY WORDS: Urban Tree Detection, Deep Learning, DeepForest, Poisson Disk Sampling

ABSTRACT:

The distribution and quantity of trees within a city impacts issues such as urban heat islands, air quality and general city planning. Having an automatic procedure for cataloguing them can be a valuable aid for future urban planning and design. In this paper, the forestry surveying neural network, DeepForest, is utilised for tree detection in the urban environment. The study area covers District Lozenets, which is the greenest part of Sofia, Bulgaria. Three distinct approaches are implemented considering the urban vegetation context - a simple tree detection, a tree cluster detection and a mixing approach between the two based on approximation with Poisson Disk Sampling. The evaluation of the developed models, in terms of F-1 score, shows that the achieved results are comparable to the ones achieved by the original application of the DeepForest model. Due to the specifics of urban data, all models tended to achieve a higher precision but a lower recall than the original DeepForest. Conditions, such as shade from the sun, buildings or other trees, make the detection more challenging. The obtained results prove the feasibility of the proposed approaches, even with a small amount of labelled data. The tree cluster and mixed approaches have the potential to resolve part of the issues coming from the urban environment context of application.

1. INTRODUCTION

Over the last six decades, urban land has expanded dramatically, surpassing the rate of population growth (Xu et al., 2019). This rapid urbanisation focuses the attention of urban planners and city authorities on the impact of cities on climate change (Asadzadeh et al., 2022). The challenges of creating sustainable and more resilient cities necessitate research on urban green areas (Sanesi et al., 2019). Their coverage, location and growth have a measurable impact on the urban environment and quality of life in cities. For example, trees positively affect air quality in cities and help mitigate extreme events such as flooding and urban heat islands (Liu et al., 2014, Nowak, 2002). The physical and mental health of the residents is also improved by the availability of green areas (Stubbings et al., 2019).

Acquiring a large and accurate tree inventory in a city is an ongoing challenge. Common approaches include on-ground surveys with physical geopositioning of trees. However, they are limited in scope, time-consuming and prone to human error. Thus, only a few cities have continuously updated tree datasets that can be used to evaluate the changes in the urban green areas and, consequently, the changes in ecosystem services(Laumer et al., 2020). This raises the need for approaches that enable the automated creation of an urban vegetation inventory.

Many approaches have relied on remote sensing to obtain information about urban green areas. For example, very high resolution (VHR) satellite images are used for individual tree crown detection (Ardila et al., 2012, Ke and Quackenbush, 2011, Anton Kuzmin and Maltamo, 2016). LiDAR-based models can yield good results on individual trees but have the disadvantages of being a costly investment, and different algorithms' accuracy is highly dependent on the data they are applied. The algorithms' performance varies even on the same datasets (Wu et al., 2016), thus creating data transferability concerns (Vaglio Laurin et al., 2019, Aubry-Kientz et al., 2019). In recent years, the advances of deep learning models have been successfully applied for object detection tasks, including tree detection. The DeepForest model is one of the most commonly used, aiming to build a generalizable tree detection pipeline applicable to all kinds of forests (Weinstein et al., 2019). The respective pipeline consists of two steps, where the first one is pre-training an object detector network's weights on comparatively noisy, self-supervised LiDAR labels, and the second is re-training the network on more precise, hand-annotated labels. The idea is to give the detector network a wide knowledge of tree shapes extracted from a LiDAR algorithm and fine-tune the acquired initial weights on the specific dataset, thus achieving high generalization. As its neural network backbone, it uses RetinaNet, which achieved state-of-the-art performance in both speed and accuracy with its unique focal loss function prioritizing foreground objects (Lin et al., 2017).

The DeepForest model has a wide range of tree detection applications, such as olive trees (Marin et al., 2022), orchard trees (Jemaa et al., 2022), and boreal forests (L. Bennett and Boisvert, 2022). It has also been used for automatic tree indexing of cities (Petrov et al., 2021). There have been other automatic tree indexing attempts on RGB data using machine learning models. For instance, transfer learning methods are combined with more traditional object detection architectures like R-CNN, along with triangulating coordinates from ground-level images with the aid of YOLO (Velasquez-Camacho et al., 2023). There have even been non-deep learning configurations attempting crown delineation with unsupervised algorithms like k-means (Moussaid et al., 2021). However, none of the identified studies seemed to tackle the specifics of urban vegetation or target groups of trees in cities growing together.

The goal of this paper is to prove that, taking the specifics of urban data into account, tree detection models for forests can be repurposed in an urban environment. A case study is conducted by applying the DeepForest model for tree detection in the District Lozenets of Sofia, Bulgaria. Detecting trees based on remote sensing data in an urban environment comes with various challenges than in a forest. There are obstructions by objects, such as buildings and other city infrastructure dropping shade on them. Additionally, the trees tend to be spread irregularly due to

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-4-W4-2024-35-2024

typically being artificially planted. Considering these specifics, three approaches are presented in the paper. They consist of a re-trained DeepForest detector on urban data, a detector trying to locate groups of trees (later referred to as 'clusters'), as well as a mixed approach trying to estimate the locations of missed individual trees.

The remainder of the paper is structured as follows. Section 2 describes the methodology and elaborates on the approaches. Section 3 shows the results and analyzes them. Section 4 concludes the paper and outlines possible future improvements. Appendix A contains tables from additional model configurations tested throughout the study.

2. METHODOLOGY

This section is dedicated to the methodology followed for the tree detection, including a description of the study area and data, model elaboration and their accuracy assessment.

2.1 Study Area

The study area covers the District Lozenets in Sofia, Bulgaria. It is characterised by significant urban complexity due to the combination of old and newly established residential areas. The district is also known as the greenest part of Sofia and thus provides a valuable use case for the research.

When object detectors like DeepForest are repurposed, the differences in data (in this case, ones between vegetation growth in the cities and the forests) must be considered. The repurposing might involve adapting the model's training data or fine-tuning its parameters to suit the specific characteristics and challenges of the urban environment. Figure 1 shows an example of trees in an urban area, where both separate and overlapping trees can be seen.

(a)

(b)

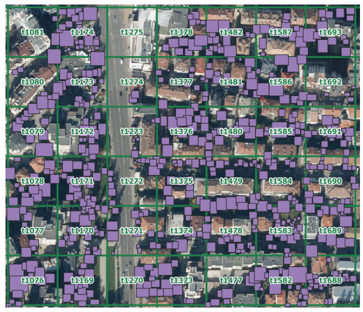
Figure 1: Trees in an urban environment: trees spread away from each other (a) and overlapping trees (b).

In addition, the trees in a city are often between large buildings, which obstruct their recognition using an RGB image. The shading and tree growth in very cramped spaces cause overlapping of tree crowns into a big cluster (Figure 1b)). Thus, the differentiation and separation of the trees in the image is difficult not only for deep-learning models but also for the human eye.

2.2 Data Acquisition and Preprocessing

The dataset used for the current study is derived from a 0.1m orthophoto image captured in 2020 using aerial photography techniques, employing an ultra-wide range digital camera (Figure 2a)). It was preprocessed to obtain tiles of 51.2m $\times$ 51.2m. Whenever a tree was between two or more tiles, it was considered a part of them. Labelling is performed on a small subset of the dataset (Figure 2b). To achieve accuracy, human-geolocated coordinates of trees were predominantly used, obtained from the EdnoDarvo organization (EdnoDarvo, 2021). Later on, the dataset was extended with easily recognisable cases such as those shown in Figure 1a). They are labelled directly using the orthophoto image. The total amount of tree images used by the end was 826.

(a)



(b)

Figure 2: Full Lozenets orthophoto (a), and labelled subset with tiles outlined in green (b).

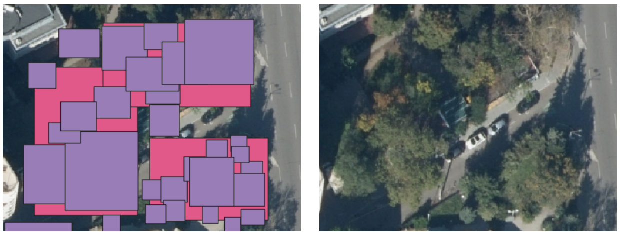
The dataset has been deliberately chosen to be close in characteristic to the images of the original National Ecological Observatory Network (NEON) dataset, which was of the same resolution and tiled into 40m $\times$ 40m (NEON, 2018) images. NEON was the dataset on which DeepForest was originally trained.

2.3 Tree Clusters

The concept of a 'tree cluster' is considered for two of the models. This study defines a tree cluster as an object consisting of two or more trees with overlapping crowns (Figure 3). The motivation for defining this object is to deal with data quality and quantity issues. Labelling the object as a cluster is simpler than differentiating between all objects within the image and can be done by a human expert directly from the remote sensing data. In total, 98 tree clusters have been labelled in the same study area as for the individual tree detection model.

2.4 DeepForest Models

The DeepForest model ensures good generalizability to different tree species and areas due to its two-stage training approach based on LiDAR and hand-annotated data. The model comes pretrained with a set of initial weights. It is trained on a very large dataset from the NEON data sites in the United States, where the labels come from a self-supervised LiDAR-based algorithm. Those labels tend to be noisy and imperfect but give the neural network a wide knowledge of tree shapes and appearances. The model can be fine-tuned for a specific site with high-quality hand-annotated data, as was done by the original study (Weinstein et al., 2019). This approach should assure good generalization to different forestry sites by simply re-training on new hand-annotated labels. Since the DeepForest model's primary purpose



(a) (b)

Figure 3: Example of how tree clusters would be labelled in relation to individual trees. Clusters are coloured pink, and individual trees are coloured purple. Original image in (a). Labelled image (b).

is forestry surveying, there is no guarantee that this will translate well to urban applications such as the ones in this study. Several approaches for dealing with this are investigated in the following subsections, considering the specifics of urban data.

2.4.1 Single Tree Detection Model: The first model is a plain re-trained version of the DeepForest model on images with single labelled trees. The goal is to detect all trees on an input tile by drawing bounding boxes around them.

2.4.2 Tree Cluster Detection Model: A tree cluster detection model disregards the exact location of individual trees in favour of estimating the location of groups of trees with overlapping crowns.

Intuitively, it supposes that the overlapping tree crowns look like a single object as in the case of a single-tree detector (Figure 1b)). DeepForest has previously demonstrated the capability to detect different tree species.

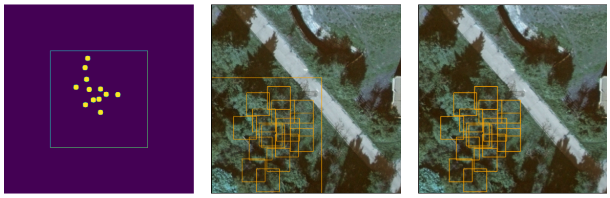
The goal of using the cluster tree dataset is to check if the model can efficiently detect groups of trees.

2.4.3 Mixed Model: Having two single-class object detectors, namely single-tree and cluster-tree detectors, naturally leads to trying to combine them. At first, there was an attempt to create a multi-class object detector before shifting to an approximation approach mixing converting clusters to individual trees.

The first approach, for a multi-class detector, attempted to combine the cluster and individual tree single-class datasets into one multi-class dataset. Unfortunately, early results on this detector were poor, seemingly due to the incompatibility of the DeepForest detector and RetinaNet architecture. As is visible in Figure 3, clusters and individual trees would overlap, which is also most likely why the results were poor. Retraining DeepForest on this dataset led to almost no objects being detected, hence this approach being dropped.

The second approach, combining the two single-class detectors, uses a heuristic-based mixing approach. It uses single and cluster tree detectors as input to create predictions. Then, the Poisson Disk Sampling algorithm is applied to convert detected clusters into separate trees while considering local tree sizes and positions and always prioritizing individual trees detected. It is the final approach considered in the evaluation.

Poisson Disk Sampling is a blue noise generation technique based on all existing points in a grid (Bridson, 2007). Points are spread a distance $[r, 2r]$ around each other, intuitively creating a pattern similar to 'dart throwing'. The version implemented works in $O(n)$ time. An illustration of how the algorithm has been used in the study context is shown in Figure 4.



(a) (b) (c)

Figure 4: Poisson Disk Sampling. Blue noise generation in a box as a grid with $k = 12$ points (a). The outline of the cluster and the approximated trees (b). The generated points in the mixed model only (c).

The proposed mixed model works as follows. The two previously described detectors process an input image and output corresponding single trees and tree clusters inside it. Then, based on the individual trees detected (if any), the average tree bounding box size is computed, and from it, a radius r is derived. The radius r is then used to spread k individual trees inside of the cluster area, which acts as the grid for the Poisson Disk Sampling algorithm. The number k is computed from the tree cluster area and the local tree size (based on r). If any individual trees are detected inside the cluster, they are used as part of the k points in the algorithm. Thus, if the tree cluster detected is accurate, the approximation points can only be false positives and lead to an overapproximation rather than an underestimation of the number of trees in the area. If tree inventory should be constructed, this can be a perfectly viable way.

2.4.4 Accuracy Assessment: The labelled data was split into training, test, and validation sets according to machine learning practices. The proportion chosen for the training-test and training validation sets was a standard of 80-20.

A sum between focal loss classification and bounding box regression loss is used for the loss function. This is the same function RetinaNet and DeepForest are optimized on. Internally, a score threshold of 0.4 is used to pre-filter bad predictions, once again a leftover from DeepForest.

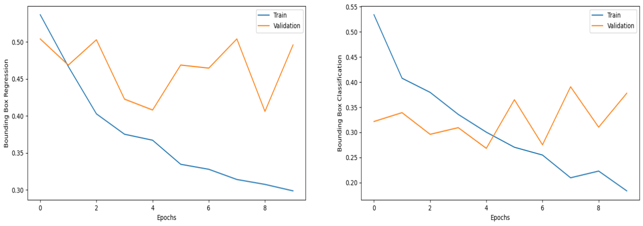
The single-tree and cluster detectors were trained for 10 epochs while monitoring their validation set loss to pick the best ones. Due to the way it is constructed, the mixed model had no explicit training step and instead had two pre-trained models as input. To mix and match different configurations, the individual tree and cluster models were trained for 1, 5 and 10 epochs and examined on the test set.

All models had their performance evaluated in the same way. As this is an object detection task, a classification rule was needed when a single tree or tree cluster was 'detected'. Achieving an IoU (Intersection over Union) over 50% on a ground truth bounding box was considered a true positive. This aligns with other object detectors, including the original DeepForest model. Such classification rule then allows using different object detection accuracy metrics, such as precision, recall, and F-1 score. Additionally, a track was kept per model of the average tree error (per tile) and the total number of trees missed (cumulative error).

3. RESULTS

The results from the best-performing configurations for each model are listed in Table 1, Table 2, Table 3 and Table 4. It should be

(a)



(b)

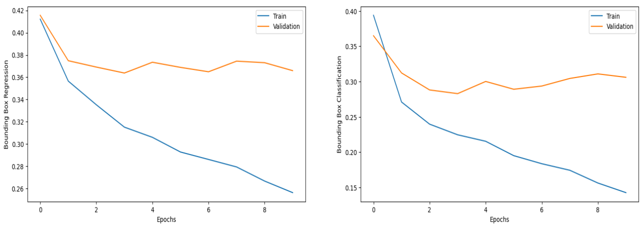
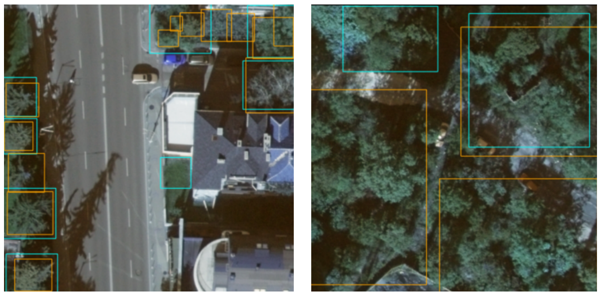


Figure 5: Training and validation set loss functions over 10 epochs for the single tree detector (a) and the cluster tree detector (b).

noted that due to the approximation nature of the Poisson Disk Sampling, subsequent runs of the mixed model can yield slightly different results. The evaluation of all configurations has been done on the test set after retraining together with the validation set.

A visualization of the training loss is provided in Figure 5. Both the single tree detector and the cluster detector had their loss presented for 10 epochs to select the best ones. For the single-tree detector, validation loss plateaued after one epoch, and for the cluster detector after five. As mentioned in the Methodology section, to evaluate the performance of the mixed model, multiple single-tree and cluster configurations are used for training and examined together. Details are discussed as follows.

For single-tree detection, the best-performing setup on the validation and test sets was with a single epoch. In the other model configurations, the model would overfit at least for single tree detection, and validation set performance would go down. Surprisingly, re-training on the full dataset did not improve the results. It decreased the F-1 score despite achieving a higher precision. The implication of this would be that not all labels in the dataset are of the same quality, and this is something to be considered for future iterations. Visual inspection of the test results showed that the single tree detector easily manages to identify single trees around streets or buildings but struggles whenever the tree crowns overlap. The most common mistake is counting two or more small trees as one, as shown in Figure 6a).



(a) (b)

Figure 6: Ground truth in orange bounding boxes and predictions in the blue ones. The single tree detector counts two or more trees as one (a). The cluster detector struggles to find all clusters in the image, counting in the top left a single tree as a cluster because of its size (b).

The best-performing cluster detector was the one trained on 5 epochs. All cluster configurations had high variability regarding each other. This can be caused by the cluster dataset having fewer objects in its total. As such, finding or missing a single cluster has a larger impact on the score, meaning subsequent epochs substantially change the results, and lead to an unstable setup. Retraining on the full dataset showed an increase in recall but a larger decrease in precision. However, the F-1 Score was the best on the test set for tree clusters. Results on the cluster model were mixed. Sometimes, it completely missed some clusters or sometimes even recognised larger single trees as clusters, as illustrated in Figure 6b). However, it also managed to detect some cases where the single tree detector missed (Figure 7), while being very accurate whenever it managed to predict. Overall, the tree cluster detection model is precise, but it struggles sometimes with identifying groups of trees as objects at all. The issue can be addressed with a larger dataset to stabilize training.

(a) (b)

Figure 7: A showcase of how the single tree detector struggles with trees that are very close together (a). A simpler task it accomplished successfully by cluster detector (b).

Two mixed model configurations show the best results. The first configuration was based on the assumption that the two best single-tree detectors would result in the best-mixed detection model and is therefore made from the 1-epoch single-tree and 5-epoch cluster detectors. However, in terms of F-1 Score on the test set, it was outperformed by another combination of mixing two 1-epoch detectors. On the full dataset retraining, however, the first mixed model performed only marginally worse than the second. This again implies that some annotations are of lower quality than others and that a larger dataset is needed, as adding such a small amount of samples should not make a big impact. Visual examination of mixing the predictions from the two detectors showed satisfactory results. The approximation samples did not go out of the predicted tree area and did not overlap with the existing trees already detected (Figure 8). Additionally, both configurations had a lower average and cumulative error than the single tree detectors across the board. If the cluster tree detection could be more consistent, this could be a perfectly viable way of mapping green areas in a city.

Figure 8: An example of trees added through approximations, visible in the top right corner. Orange bounding boxes represent ground truth, and the blue ones predictions.

For forestry surveying, the original DeepForest model achieved an average tree crown recall of 0.69, a precision of 0.61, and a resulting F-1 score of 0.65. For all detectors and their configurations, the results in an urban environment are comparable and show the potential in the model's reuse for an urban environment.

Set	Precision	Recall	F1-Score	Avg. Err	Cum. Err
Training	0.762	0.623	0.685	-3.154 trees	-104/497 trees
Test	0.794	0.594	0.680	-3.909 trees	-43/185 trees
Full	0.821	0.485	0.610	-7.454 trees	-82/185 trees

Table 1: Single Tree Detector-1 Epoch

Set	Precision	Recall	F1-Score	Avg. Err	Cum. Err
Training	0.661	0.524	0.585	-0.714 clusters	-20/64 clusters
Test	0.700	0.278	0.398	-1.555 clusters	-14/21 clusters
Full	0.5208	0.426	0.468	-0.888 clusters	-8/21 clusters

Table 2: Cluster Detector-5 Epochs

Set	Precision	Recall	F1-Score	Avg. Err	Cum. Err
Training	0.660	0.629	0.644	0.393 trees	13/497 trees
Test	0.779	0.594	0.674	-3.636 trees	-40/185 trees
Full	0.772	0.4931	0.601	-5.909 trees	-65/185 trees

Table 3: Mixed Tree Detector-1 Epoch

Set	Precision	Recall	F1-Score	Avg. Err	Cum. Err
Training	0.631	0.579	0.604	5.78 trees	191/497 trees
Test	0.719	0.599	0.654	0 trees	0/185 trees
Full	0.767	0.487	0.596	-5.909 trees	-67/185 trees

Table 4: Mixed Model with Best Test Performance - Single - 1 Epoch & Cluster - 5 Epochs

4. CONCLUSIONS AND FUTURE WORK

The application of the DeepForest model for tree detection on an urban dataset is successful, even with the lack of labelled data. The best-performing single tree detection models and mixed models managed to achieve results comparable to the original DeepForest model according to the F-1 Score. The current study handles the challenges of the urban environment, where it is hard to detect trees obstructed by buildings in shades or overlapping with each other. Nevertheless, using a comparatively small dataset, the proposed urban tree detection solution achieved satisfactory performance while not breaking the generalization approach of the DeepForest detector or overfitting to the dataset. The approximation approach mixing tree clusters and single trees has potential for application in cases where labelled data is sparse. Visual examinations of the results and the increase in recall show that overapproximating more trees is not an issue.

Further experiments and possibilities for improvement are identified. The issues with the data quality need to be addressed, which requires repeating the experiment on different datasets with far more labels. A data augmentation could be applied to the current dataset to create more tiles from intersections between them.

Certain parameter adjustments and independent variables were not isolated during the study. The models were not tested on negative samples (explicitly testing images without trees in them), as early results proved a technical challenge and a hindrance. To ensure applicability on a larger scale, the model configurations should be verified so as not to detect trees in places where there are none or expanded to work with negative samples. Possible options would be explicitly adding negative samples to the dataset or filtering them out through post-processing using the NDVI for a given image.

During some experiments with the mixed model, it was observed that some of the approximations in a cluster were too small compared to the local trees in the area. This is because whenever no other trees were detected in the dataset, a default parameter was used, simply the average tree area over the entire dataset. Future improvements could include making a more dynamic default size selection or extending the locality of the trees beyond just one image. More experiments should be conducted on a multi-class detector. As was discussed in the Results section, RetinaNet was not directly suitable, but that does not mean another neural network architecture cannot be. An example would be a more traditional R-CNN. If the tree clusters could be a part of an object detector network without approximation, then results would be more consistent while keeping the benefits of the simplified annotation.

Finally, precision and recall may not be the best training metrics for the approximation mixed model. However, they are not ideally suited for approximation approaches, which are inexact by nature. Precision is hindered, as it is hard to reach the 50% IoU criterion discussed previously is hardy. As it is now, results that are 'good enough' visually end up severely punished in the automatic evaluation.

In conclusion, tree detection from remote sensing data is challenging, where taking the data specifics into account helps. The difficulties come from sparse data and the labelling complexity, even from human experts. Developing a general approach that handles urban environment specifics enables an efficient inventory of trees and easy creation of 3D city models, following standards like CityGML. Although positive results are being achieved, there is room for further investigation, considering the lessons learned from the pitfalls presented in this paper.

ACKNOWLEDGEMENTS

This research is part of the GATE project supported by the Horizon 2020 WIDESPREAD-2018-2020 TEAMING Phase 2 programme under Grant Agreement No. 857155 and the enRichMyData project supported by the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101070284. The authors gratefully acknowledge the support of the FLEdge project, Agreement No. KP-06-D002/5, funded by the European partnership Driving Urban Transitions (DUT), cofunded by the European Commission.

REFERENCES

Anton Kuzmin, Lauri Korhonen, T. M. and Maltamo, M., 2016. Automatic segment-level tree species recognition using high resolution aerial winter imagery. European Journal of Remote Sensing 49(1), pp. 239–259.

Ardila, J. P., Bijker, W., Tolpekin, V. A. and Stein, A., 2012. Context-sensitive extraction of tree crown objects in urban areas using vhr satellite images. International Journal of Applied Earth Observation and Geoinformation 15, pp. 57–69. Special Issue on Geographic Object-based Image Analysis: GEOBIA.

Asadzadeh, A., Kötter, T., Fekete, A., Moghadas, M., Alizadeh, M., Zebardast, E., Weiss, D., Basirat, M. and Hutter, G., 2022. Urbanization, migration, and the challenges of resilience thinking in urban planning: Insights from two contrasting planning systems in germany and iran. Cities 125, pp. 103642.

Aubry-Kientz, M., Dutrieux, R., Ferraz, A., Saatchi, S., Hamraz, H., Williams, J., Coomes, D., Piboule, A. and Vincent, G., 2019. A comparative assessment of the performance of individual tree crowns delineation algorithms from als data in tropical forests. Remote Sensing.

Bridson, R., 2007. Fast poisson disk sampling in arbitrary dimensions.

EdnoDarvo, 2021. EdnoDarvo - Online Dataset of Tree Canopies in Sofia, Bulgaria.

Jemaa, H., Bouachir, W., Leblon, B. and Bouguila, N., 2022. Computer vision system for detecting orchard trees from uav images. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLIII-B4-2022, pp. 661–668.

Ke, Y. and Quackenbush, L. J., 2011. A review of methods for automatic individual tree-crown detection and delineation from passive remote sensing. International Journal of Remote Sensing 32(17), pp. 4725–4747.

L. Bennett, B. Wilson, S. S. L. Q. M. W. H. Z. and Boisvert, J., 2022. Image to attribute model for trees (itam-t): individual tree detection and classification in alberta boreal forest for wildland fire fuel characterization. International Journal of Remote Sensing 43(5), pp. 1848–1880.

Laumer, D., Lang, N., van Doorn, N., Mac Aodha, O., Perona, P. and Wegner, J. D., 2020. Geocoding of trees from street addresses and street-level images. ISPRS Journal of Photogrammetry and Remote Sensing 162, pp. 125–136.

Lin, T.-Y., Goyal, P., Girshick, R., He, K. and Dollár, P., 2017. Focal loss for dense object detection. pp. 2999–3007.

Liu, W., Chen, W. and Peng, C., 2014. Assessing the effectiveness of green infrastructures on urban flooding reduction: A community scale study. Ecological Modelling 291, pp. 6–14.

Marin, I., Gotovac, S. and Papić, V., 2022. Individual olive tree detection in rgb images. In: 2022 International Conference on Software, Telecommunications and Computer Networks (SoftCOM), pp. 1–6.

Moussaid, A., Fkihi, S. E. and Zennayi, Y., 2021. Tree crowns segmentation and classification in overlapping orchards based on satellite images and unsupervised learning algorithms. Journal of Imaging.

NEON, 2018. San Joaquin Experimental Range - National Ecological Observatory Network Dataset.

Nowak, D. J., 2002. The effects of urban trees on air quality.

Petrov, A., Slavkova, A., Rahia, Y., Hristov, K., Mechkova, L., Borukova, M., Stoykov, N. and Malinov, S., 2021. Filing of durable vegetation.

Sanesi, G., Giannico, V., Elia, M. and Lafortezza, R., 2019. Remote sensing of urban forests. Remote Sensing.

Stubbings, P., Peskett, J., Rowe, F. and Arribas-Bel, D., 2019. A hierarchical urban forest index using street-level imagery and deep learning. Remote Sensing.

Vaglio Laurin, G., Ding, J., Disney, M., Bartholomeus, H., Herold, M., Papale, D. and Valentini, R., 2019. Tree height in tropical forest as measured by different ground, proximal, and remote sensing instruments, and impacts on above ground biomass estimates. International Journal of Applied Earth Observation and Geoinformation 82, pp. 101899.

Velasquez-Camacho, L., Etxegarai, M. and de Miguel, S., 2023. Implementing deep learning algorithms for urban tree detection and geolocation with high-resolution aerial, satellite, and ground-level images. Computers, Environment and Urban Systems 105, pp. 102025.

Weinstein, B. G., Marconi, S., Bohlman, S., Zare, A. and White, E., 2019. Individual tree-crown detection in rgb imagery using semi-supervised deep learning neural networks. Remote Sensing.

Wu, B., Yu, B., Wu, Q., Huang, Y., Chen, Z. and Wu, J., 2016. Individual tree crown delineation using localized contour tree method and airborne lidar data in coniferous forests. International Journal of Applied Earth Observation and Geoinformation 52, pp. 82–94.

Xu, G., Jiao, L., Yuan, M., Dong, T., Zhang, B. and Du, C., 2019. How does urban population density decline over time? an exponential model for chinese cities with international comparisons. Landscape and Urban Planning 183, pp. 59–67.

APPENDIX A: FULL TABLES

This appendix contains all available tables from all other training configurations explored during the experiment. The tables here feature no retraining on the full training set. All results have been rounded down to three digits after the decimal point.

Set	Precision	Recall	F1-Score	Average Error	Cumulative Error
Training	0.820	0.600	0.693	-4.666 trees	-154/497 trees
Test	0.764	0.496	0.601	-6.181 trees	-68/185 trees

Table 5: Single Tree Detector-5 Epochs

Set	Precision	Recall	F1-Score	Average Error	Cumulative Error
Training	0.884	0.760	0.817	-2.696 trees	-89/497 trees
Test	0.694	0.573	0.628	-3.727 trees	-41/185 trees

Table 6: Single Tree Detector-10 Epochs

Set	Precision	Recall	F1-Score	Average Error	Cumulative Error
Training	0.508	0.276	0.358	-1.107trees	-31/64 trees
Test	0.833	0.204	0.327	-1.888 trees	-17/21 trees

Table 7: Cluster Detector-1 Epoch

Set	Precision	Recall	F1-Score	Average Error	Cumulative Error
Training	0.970	0.767	0.857	-0.678 trees	-19/64 trees
Test	0.500	0.185	0.270	-1.666 trees	-15/21 trees

Table 8: Cluster Detector-10 Epochs

Set	Precision	Recall	F1-Score	Average Error	Cumulative Error
Training	0.660	0.611	0.635	0.090 trees	3/497 trees
Test	0.705	0.504	0.588	-3.72 trees	-41/185 trees

Table 9: Mixed Tree Detector-5 Epochs

Set	Precision	Recall	F1-Score	Average Error	Cumulative Error
Training	0.711	0.763	0.736	2.545 trees	84/497 trees
Test	0.641	0.581	0.609	-1.090 trees	-12/185 trees

Table 10: Mixed Tree Detector-10 Epochs