Urban Tree Cover Segmentation from High-resolution Aerial Imagery of Göttingen

Rohit Yadav, Amit Bharti, Pauline Drews, Esther Cros

Abstract—Satellite imagery has become an important source of data for forest management, providing valuable insights into the state of forests at a large scale. However, analyzing these images manually can be a time-consuming and challenging task. In recent years, deep learning techniques have shown great promise in automating image analysis tasks such as semantic segmentation. In this study experiments were conducted using deep learning-based architectures for semantic segmentation of urban satellite images of Göttingen. The experimental results show that model outperforms other models in terms of segmentation accuracy, with an average intersection over union (IoU) score of 0.79. This approach could be a valuable tool for forest management, enabling more efficient analysis and monitoring of forests using satellite imagery. The implementation is available here.

Index Terms—Forestry, Deep learning, Transfer learning, Tree cover segmentation, U-Net, DeepLabV3.

1 Introduction

Rapid worldwide urbanization is one of the strongest drivers for the environmental degradation of urban regions [2]. Average daily increases of settlement and traffic area in Germany were estimated at 55 ha/d from 2018 to 2021 without a clear decreasing trend [39]. The main effects of this degradation are greater noise exposure, rising greenhouse gas emissions, air and water pollution, erosion, and biodiversity loss [47, 44, 20].

Urban trees can mitigate the effects of environmental degradation through various ecosystem services and further benefits. Ecosystem services are "specific results of ecosystem functions or aspects of ecosystems utilized actively or passively, directly or indirectly, to sustain or enhance human and non-human life" [15] and have four scopes: provisioning, regulation, supporting, and cultural services [8], of which the regulating and cultural services are most relevant in urban regions [5, 9]. Some important regulating services of urban trees are cooling and reduction of heat islands, energy conservation, stormwater attenuation and flood risk reduction, air filtration, noise reduction, carbon sequestration, and wind reduction [30, 17, 28, 5]. Essential cultural services of urban trees are the improvement of aesthetics, nature experience for citizens, and reduced crime rates [14, 24, 26]. Furthermore, urban trees have a positive influence on mental and physical health [25, 41] and an economic value, both directly, e.g. by increasing property values [4], and indirectly by preventing costs that would incur for the regulating services if there were no trees [36].

Climate change endangers urban trees and their various services and benefits. The fast process of changing climatic conditions changes the living circumstances faster than the trees are able to adapt, especially in an urban region, where the effects occur earlier and are stronger. The resulting physiological stress makes the trees more prone to pests and pathogens and increases their mortality probability [42]. To both quantify the services of urban trees and guarantee an early detection of negative trends, precise and temporal explicit monitoring of urban tree cover is needed [13]. Knowledge of the tree cover then provides information on the location, distribution, size, and form of tree canopies in urban areas [43]. For this, technological approaches allowing fast data collection and processing are needed.

- *Rohit Yadav is with the University of Groningen, E-mail: r.yadav.2@rug.nl.*
- *Amit Bharti is with the University of Groningen, E-mail: a.bharti.1@student.rug.nl.*
- *Pauline Drews is with the University of Göttingen, E-mail: pauline.drews@stud.uni-goettingen.de.*
- *Esther Cros is with the University of Bordeaux, E-mail: Esther.cros@etu.u-bordeaux.fr.*

Semantic Segmentation is a useful method to assess tree cover information from high-resolution aerial images. It consists of the assignment of a label to each pixel of an image and results in a segmentation map that portrays the objects of the image and their assigned class [45]. In a tree cover segmentation task, only two classes, "tree" and "no tree", are distinguished. Traditionally, urban tree data was assessed using digital recordings in fieldwork, but rapid progress in information technology enhanced the possibilities in data gathering and analysis. With improved sensor technology, high-resolution remote sensing imagery was provided, and increasing computational processing power allowed to interpret it [6, 29, 21]. The first important approach that extracted urban structures from high-resolution images was the Object Based Image Analysis (OBIA) [7]. In addition to spectral data, it also uses information on the texture, shape, and context of objects and reaches higher accuracies than per-pixel classification techniques for small features in urban areas [32]. However, OBIA has some disadvantages, one of them being the complex process of scale parameter selection that needs domain-specific knowledge [22]. This problem can be solved with deep learning approaches which work in an end-to-end fashion, making previous expert knowledge unnecessary and at the same time reaching higher accuracies.

The current state-of-the-art deep learning approach for semantic segmentation of urban tree covers is Convolutional Neural Networks (CNN). The first CNN was developed by LeCun et al. [27] in 1998. CNNs became more popular in 2012 with the AlexNet [3] and TensorFlow [1], which made the application more accessible to scientists of application research fields. In semantic segmentation, the Fully Connected Network (FCN) of Long et al. [38] was great progress in the field, as it allowed arbitrary image sizes for segmentation tasks. In 2015, Noh et al.![31] published the Dilated Convolutional Neural Networks (DCNN) which have been improved and modified since then and are still one of the state-of-the-art networks for semantic segmentation of urban structures [29].

For semantic image segmentation, or rather a pixel-wise classification, a clear definition of the classes is needed. In our project, "tree" and "no tree" classes were predetermined by the used mask dataset. However, some general rules for the classifications of urban trees could be derived from that. Roy et al. [36] described an urban tree as "a woody perennial plant growing in towns and cities, typically having a single stem or trunk - and usually a distinct crown - growing to a considerable height, and bearing lateral branches at some height from the ground". They included individuals, just like aggregations of trees in public urban areas. As a consequence, objects like meadows, sports fields, bushes, trees outside urban areas, and trees in private spaces were not included. This definition matches well with mask dataset, with the exception of trees on private grounds, which were also included in our dataset.

Extracting features like tree covers from aerial images of urban areas includes several challenges due to the complex structure of the target environment. Urban regions are a unique, heterogeneous juxtaposition of natural and constructed elements. Thus, buildings and streets of geometric shapes are strongly nested with complex-shaped trees on a very small spatial scale. Also, trees are distributed heterogeneously over the area and a variety of species occur. This involves interferences between classes on a very small scale and in irregular patterns. Additionally, there are members of different classes with similar specters that only differ in shape or on non-visible spectrum channels, e.g. vegetation other than trees and trees. Furthermore, shadows of different sizes, forms, and shades lay over objects and increase the palette of spectral compositions that an object of a certain class can have. Lastly, label datasets are mostly generated manually and can be objects of human errors that lead to wrong masks and unideal learning conditions for the algorithm.

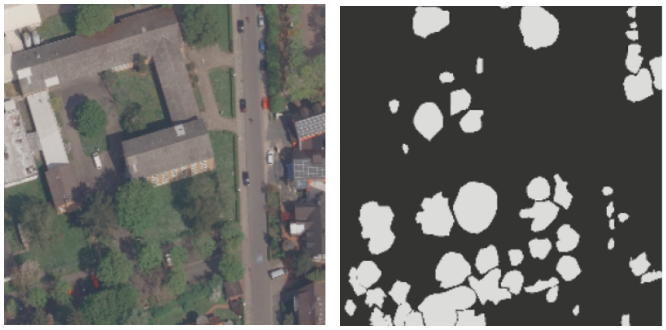
In this work, we implemented a tree cover segmentation task on a dataset of aerial images and corresponding masks of the German city Göttingen. We compared the performance of a baseline model using U-Net architecture [34] to a DeepLabV3 [10] architecture with increasingly complex ResNet architectures in the encoder.

2 Methods

In this section, we describe the methodology used to develop a forest crown segmentation model using U-Net and DeepLabV3 architectures. We explain the steps taken to preprocess satellite images, train the models using the dice loss function, and evaluate their performance.

2.1 Dataset

This study is centered on a dataset of high-resolution aerial images captured from the urban environment of Göttingen, Germany. The dataset comprises 38 images, each with a 10 cm resolution and a dimension of 1024 x 1024, captured in the red, green, blue, and near-infrared (NIR) spectral bands. In addition to the aerial images, corresponding ground truth masks are available with 2 labels: 0 for the background and 1 for the tree crown. These masks were used to train a semantic segmentation model, which could then be employed to calculate the tree cover percentage and monitor ecological changes in the urban environment over time. Fig. 1 shows one of the images from the dataset along with its mask. From Fig. 1, it could be seen that tree crowns are labeled as white patches, while green grasses and houses are labeled as backgrounds with black color pixels.



(a) Input image (b) Ground truth mask

Fig. 1. Sample image from the Göttingen dataset with its ground truth mask.

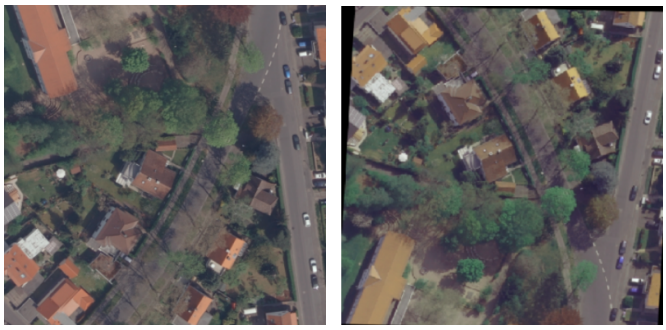
2.2 Pre-Processing

Firstly, the raw data were in Tag Image File (TIF) format having four channels, the first three were color channels namely Red, Green, and Blue (RGB) while the fourth channel was infrared. Since all the images were taken during the daylight, so infrared information from the data is expected not to play a major role in the segmentation task. Furthermore, most of the predefined architecture in computer vision takes three channels as an input, consequently trained weights on ImageNet [37] were limited to three channels only, hence choice was made to remove the infrared channel.

Secondly, the image size was reduced to 256 x 256 from 1024 x 1024 using bi-cubic interpolation over a 4 × 4 pixel neighborhood to reduce the computation power requirement and remove noise from the image. Reduction of image size was done for a few experiments, while the rest experiments were carried on with the original size to avoid information loss happening due to reduction.

Thirdly, the dataset was split into two parts which were training and testing set before training the neural network so that information leakage is avoided. Fourthly, the train and test datasets were normalized so all the pixel values were between 0 and 1.

Lastly, augmentations were applied to the training dataset before feeding them to the neural network, as discussed below.



(a) Original Image (b) Augmented Image

Fig. 2. Original image and augmented image after various augmentation techniques applied to it based on Albumentations algorithm for that instance.

2.2.1 Data Augmentation

Training a neural network efficiently requires feeding the model with a lot of data, but the current dataset has only 38 images. Hence, data augmentation could be a good alternative here to generate more training samples. Augmentation accomplishes two goals: First, more labeled training data can theoretically increase the performance of the model since now it has more samples to fine-tune the neural network parameters, but on the other hand, also lead to over-fitting. Second, the model can train on several orientations because of the transformations. This allows the model to generalize better and gives the model some wiggle room when it comes across slight variation swings in testing or real-world data, as observed by Glickman et al. [16]. Augmentation is applied using the Albumentations [1] library, which applies multiple augmentation techniques to the image and the mask simultaneously based on the probability value defined for each augmentation.

Augmentations that were applied to the images in the pipeline are as follows:

- *Gaussian Blur:* Blurs the input image using a Gaussian filter with a random kernel size.
- *Sharpen:* Sharpens the input image and overlays the result with the original image.
- *Gaussian Noise:* Apply Gaussian noise to the input image.
- *RandomBrightnessContrast:* Randomly change the brightness and contrast of the input image.
- *Elastic transform:* Deformations applied on input images through shear and deformation.
- *Grid Distortion:* Distorts input image on set sizes of grids.

[1] https://github.com/albumentations-team/albumentations

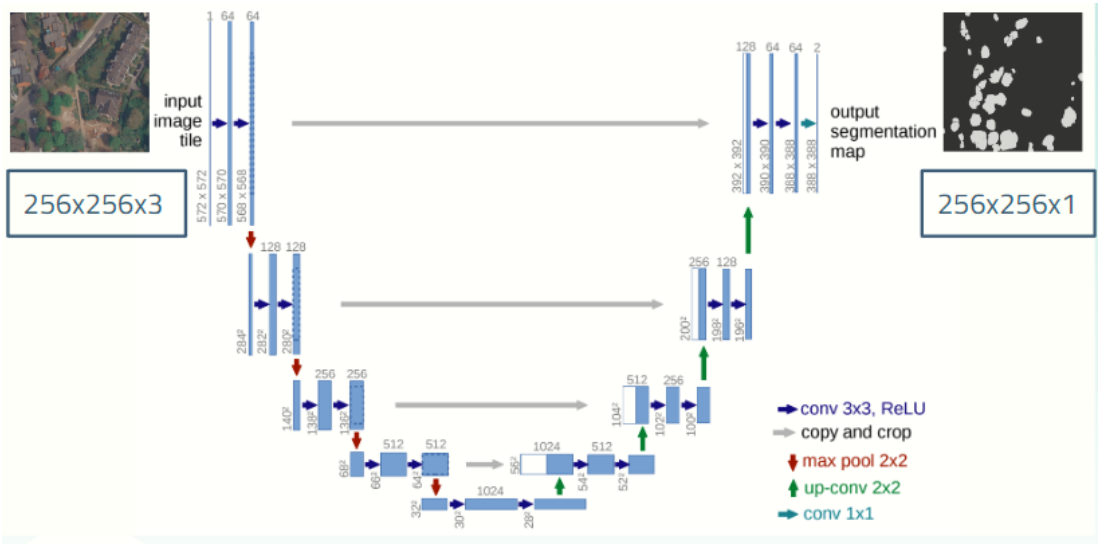


Fig. 3. The U-Net architecture is a convolutional neural network used for semantic segmentation. This figure shows its U-shaped architecture with a contracting path for encoding and an expanding path for decoding that produces pixel-wise segmentation maps. Image taken from [35].

- *Flip:* Flip the input image along the x-axis or y-axis.
- *Optical distortion:* Applies a divergence from rectilinear projection to the input image.
- *ShiftScaleRotate:* Randomly apply affine transforms: translate, scale, and rotate the input.
- *Color Jitter:* Randomly changes the brightness, contrast, and saturation of an image.
- *Down Sample:* Decreases image quality by downscaling and upscaling back.
- *RGB shift:* Randomly shift values for each channel of the input RGB image.

Fig. 2 (a) shows the original image, while Fig. 2 (b) shows the augmented image of the same. It can be seen that multiple augmentation techniques were applied to the original images. From Fig. 2, it could be seen that vertical flip, RGB shift, and elastic transform were applied by the Albumentations algorithm on that example.

2.3 Model Framework

Neural networks were trained using two different frameworks, namely U-Net and DeepLabV3, to perform the segmentation of trees. Both of them use an encoder and decoder for training the network. The encoder performs the contraction where feature information (channels) is increased while the spatial information is reduced (size of image), whereas in the decoder vice versa takes place to reconstruct the output image.

2.3.1 U-Net

U-Net [35] is used to perform the segmentation task on an input image. Since we have a limited training dataset, this architecture suits best for our use case as it has fewer trainable parameters. U-Net uses multiple blocks of convolution and max pooling to learn features of the image, so accurate segmentation of the image can take place. In each block, it has two layers of convolution each followed by batch normalization, then max-pooling is applied to lower the dimension of the image by half followed by dropout. Batch normalization and dropouts were used as a regularization technique to avoid the model from overfitting. The general architecture of the U-Net is shown in Fig. 3.

Kernel size for convolution is 3 x 3 (with "identical" padding), each followed by a non-linear ReLU activation function, while for max pooling 2 x 2 kernel size of stride 2 is chosen. Since the input image is 256 x 256, we used three blocks in total as encoder and decoder so at the end of the encoder block, we have an array size of 32 x 32 with 128 channels. In the code block, two layers of convolution were performed to increase the channels to 256. In the decode block, vice versa takes place, in addition to the combination of the spatial information and the features through a sequence of up-convolutions and concatenations with high-resolution features from the encoder block.

2.3.2 DeepLabV3

DeepLapV3 [11] is one of the state-of-the-art methods for semantic image segmentation developed by researchers at Google. It is widely used in autonomous driving, medical imaging, and tree segmentation on aerial images. It uses 2 types of neural networks which have spatial pyramid pooling which captures rich contextual information by pooling features at different resolutions and an encoder-decoder architecture to obtain sharp object boundaries. The resolution of the extracted encoder features is controlled by the atrous convolution to tradeoff between precision and run time.

In the DeepLabV3 decoder, Atrous Spatial Pyramid Pooling (ASPP) block was used, inspired bu Chen et al. [11]. This was a modification of earlier versions of the DeepLab decoder, where atrous convolution layers were used instead of normal convolution layers. Atrous convolution allows the network to efficiently increase the receptive field of the convolutional layers, without increasing the number of parameters. This means that the network is able to capture more contextual information in an image, without having to sacrifice computational efficiency. In this ASPP block, there were four atrous convolution layers, which perform convolution operations with different dilation rates to capture multiscale contextual information. Additionally, ASPP has an average pooling layer that was applied to the feature maps from the encoder block to provide global context information. The outputs of all these five layers of the ASPP block were concatenated and bilinear upsampling was applied to produce feature maps with the same dimensions as that of the input image. General architecture of the DeepLabV3 is shown in Fig. 4

2.3.3 Comparison of the Two Frameworks

Both U-Net and DeepLabV3 are CNN-based architectures commonly used for image segmentation. The objective of both networks is to segment images into individual pixels and assign them to specific categories.

One of the main differences between the two architectures is their approach to spatial resolution. U-Net is designed to integrate information at different spatial scales by using skip connections to combine information from different network layers. U-Net thus allows features

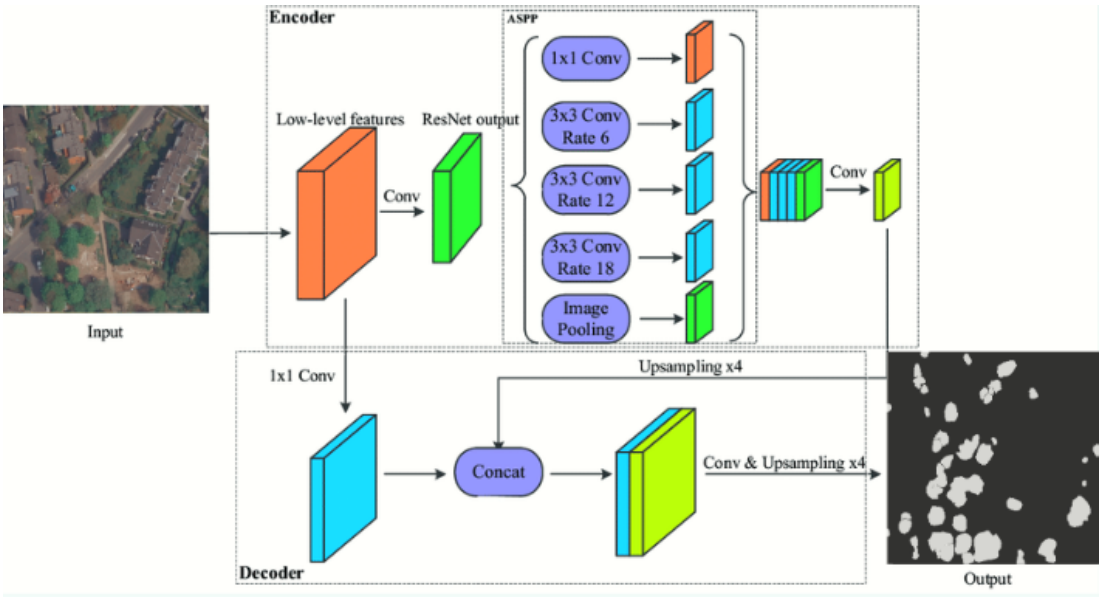


Fig. 4. The DeepLabV3 architecture is a convolutional neural network used for semantic segmentation. This figure shows an input RGB image fed into the encoder, and features learned from it are passed to the decoder, which finally produces pixel-wise segmentation maps. Image taken from [46].

to be captured at different spatial resolutions. It allows accurate results to be obtained even with a small training dataset. In this study, we only had 30 training images, which is very small. DeepLabV3 uses dilated convolutions to increase the receptive field of each layer of the array. It allows the capture of large-scale contextual information without increasing the number of parameters. This technique improves segmentation accuracy by allowing the network to better understand the overall image structure.

Also, DeepLabV3 is generally considered more accurate than U-Net on large images at the expense of higher complexity and longer processing time. U-Net, on the other hand, is generally faster and lighter in terms of complexity but can have accuracy problems when used on large images.

2.4 Transfer Learning

The size of the training dataset is a critical factor in deep learning, as it influences the capacity of the model to learn the underlying patterns and features present in the data. With only 38 images available for training, it is likely that the model will not be able to capture all the relevant information in the data, leading to poor performance and even worse generalization to novel data. Transfer learning is a popular technique in deep learning to overcome small dataset sizes, which involves taking a pre-trained model and fine-tuning it for a specific task or dataset. A pre-trained neural network on a large dataset (such as ImageNet [37]) acts as a starting point for training a new model on a different but related dataset. By doing so, the model can take advantage of the rich feature representations learned from the large dataset, which can significantly improve the performance on the downstream segmentation task.

The high-level features learned by the pre-trained model (such as edges, textures, and shapes) are useful for the main task, even if the lower-level features (such as object segmentation) are different. A pre-trained model also avoids the need to train a new model from scratch, which can be time-consuming and resource-intensive. One approach in transfer learning is freezing the weights of the pre-trained layers and only training new layers on top of them. Another approach is to fine-tune the entire pre-trained model, where we adjust the weights of all layers in the model for the new task.

2.5 Loss Function

A loss function is used to compare the ground truth and predicted output values. It measures how well the model learns from the training data, so the aim of the neural network is to reduce the disparity between the ground truth and predicted outputs. DSC (Dice Similarity coefficient) [18] loss is used as the loss function, for which the formulas are given below:

$$DSC\ loss = 1 - DSC \tag{1}$$

$$DSC\ loss = 1 - \frac{2 * True\ Positive}{2 * True\ Positive + False\ Positive + False\ Negative} \tag{2}$$

According to the above equation, DSC is two times the area of overlap between the predicted and the ground truth divided by the total number of pixels in both images. Hence, the more the overlap between the predicted and the ground truth pixel, the more will be the DSC. The range of DSC is [0,1], where 0 means no overlap while 1 means complete overlap between predicted and ground truth labels.

2.6 Evaluation Metric

The Intersection over Union (IoU) is used as the evaluation parameter for assessing the model's performance. It is one of the most commonly used metrics for image segmentation tasks to evaluate the performance of two models [33]. IoU measures the extent of overlap between the ground truth and predicted output. The greater the overlap, the higher will be the IoU score. Similar to the DSC score, the range of IoU is [0,1], where 0 means no overlap while 1 means complete overlap between predicted and ground truth labels. The mathematical formula for IoU is given below:

$$IoU = \frac{Area\ of\ intersection\ between\ ground\ truth\ and\ prediction}{Area\ of\ union\ between\ ground\ truth\ and\ prediction} \tag{3}$$

3 Experimental Setup

Encoder: As explained in Sec. 2.3, an encoder is responsible for extracting high-level features from the input image and encoding them into a compact representation that can be easily decoded by the decoder to generate a segmentation map.

Fig. 5. Baseline experiment training and validation IoU score and DSC loss curve.

Fig. 6. Best DeepLabV3 model experiment results with transfer learning displaying the IoU score and DSC loss curve for training and testing dataset.

The following encoders were used: U-Net [35], Xception [12], as well as ResNet [19] encoder with depths of 18, 50, 101, and 152.

Decoder: The decoder is responsible for reconstructing the high-resolution segmentation map from the low-resolution feature maps produced by the encoder. Two decoder architectures were used, as described in Sec. 2.3 namely U-Net and DeepLabV3.

Training: Segmentation models were trained using two deep learning frameworks. Models from scratch were trained using TensorFlow, while the models using transfer learning techniques were trained on PyTorch. While training, batch size of 2,4 and 8 were used with Adam (Adaptive Moment Estimation) [23] as an optimizer with learning rate = 0.0001. Adam is the most popular and has proven to perform better in most deep learning tasks since it computes adaptive learning rates for each parameter considering both the decaying average of the past gradient and the decaying average of the past squared gradients. Early Stopping was used while training the model to avoid overfitting on train dataset. A significant drop in validation loss was seen before saving/updating the saved model.

Computational Power Usage: Models from scratch were trained using TensorFlow-GPU (Version 2.11) having Python 3.9 as the programming language. In addition, CUDA 12 (for GPU computing) was used for accelerated training on the NVIDIA GeForce GTX 1050 Ti. Models with transfer learning technique used PyTorch as the deep learning framework, which were trained using a Tesla K40 GPU for approximately four hours/each on the Google Colab cluster.

4 Results & Discussion

In this section, the outcomes and analysis of a series of experiments were conducted using various encoder-decoder combinations are presented, along with an examination of the impact of utilizing transfer learning on the findings. During the experiments, different encoder-decoder combinations were utilized to observe the effect of various model architectures on the results. The results of each combination were then analyzed to determine which combinations produced the most accurate and effective segmentation maps.

Fig. 7. Input image with ground truth and predicted label masked on sample input image.

4.1 Baseline model

The encoder and decoder architecture of U-Net were utilized for training the model, which was trained from scratch. The IoU and DSC loss plot is depicted in Fig.5. As the number of epochs increase, there is a clear trend of improvement in both the IoU scores and DSC loss. The IoU scores, which measure the overlap between the predicted segmentation mask and the ground truth mask, consistently increased over time, indicating that the model was becoming more accurate in its predictions. Similarly, the DSC loss, which measure the dissimilarity between the predicted and ground truth segmentation masks, consistently decreased, indicating that the model was producing more precise segmentation masks. Around 200 epochs, the baseline model achieved an IoU score of 0.6973 and a DSC loss of 0.1790. Early stopping with patience 10 was used to avoid the model from overfitting. These results are highly encouraging and suggest that the model is capable of accurate and precise image segmentation even though the model was trained from scratch on a very small dataset.

4.2 Transfer Learning

Based on the initial experiments, it was observed that the baseline model was able to produce satisfactory results when trained from scratch. However, in an effort to further enhance the performance of the model, transfer learning technique was explored.

A transfer learning approach was implemented by utilizing various encoders that had been pre-trained on the ImageNet dataset. Xception and Resnet50 were used as the encoders, while keeping the U-Net decoder unchanged. Furthermore, the performance of the baseline U-Net was compared with the DeepLabV3 decoder architecture with different encoders. To achieve this, ResNet encoder was utilized with depths of 18, 50, 101, and 152, all of which had been pre-trained on the ImageNet dataset. The IoU score of U-Net and DeepLabeV3 decoder using different encoders are given in Fig. 9.

Model	Encoder/ Decoder	Params, M	DSC Loss	IoU Score
Baseline	U-Net/ U-Net	1.9M	0.1790	0.6973
Best	Resnet152/ DeepLabV3	61.3M	**0.1281**	**0.7989**

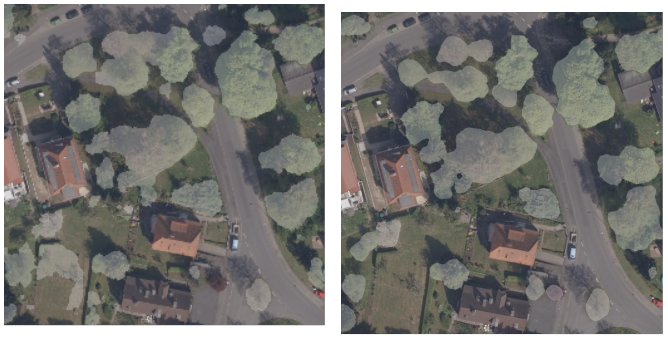
Table 1. Comparison of the performance of the baseline and top performing model

The results of our analysis demonstrate that incorporating transfer learning into the segmentation model leads to a marked improvement in the quality of the final segmentation map. Experimental findings reveal that the use of DeepLabV3 as the decoder plays a critical role in achieving the enhancement. Furthermore, experimental results reveal that the selection of an appropriate encoder has a notable impact on the performance of the segmentation model. Meticulous investigation indicates that the best results were obtained using Resnet152 as encoder, which was pre-trained on the ImageNet dataset, in conjunction with the DeepLabV3 decoder. This configuration achieved an IoU score of 0.7989 and a DSC loss of 0.1281. The performance of the best model, as measured by its training IoU accuracy and DSC loss, is illustrated in Fig. 6.

Moreover, the study also highlights that using deeper encoders leads to a more impressive performance of the model, as visible from Fig. 9. One of the reasons behind might be the large image size.

4.3 Baseline Vs Transfer Learning

The performance of the baseline method with U-Net encoder-decoder was compared with the best model, which is Resnet152 encoder with pre-trained weights and DeepLabV3 decoder. Firstly, quantitative comparison between these models were done using the number of trainable parameters, IoU score, and DSC loss as stated in Table 1. Although the high performance of the best model is apparent, one important observation to make is the difference between the parameters of the baseline and the best method. The parameters of the baseline method were, ≈ 1 %, of the total parameters of the best model. But the IoU score of the best model is only increased by 0.1016, which is a 14.57% improvement. These results indicate that it is possible to attain satisfactory results without transfer learning, but they can be



(a) (b)

Fig. 8. Resulting segmentation map of the baseline U-Net model and best DeepLabV3 model on the same input image.

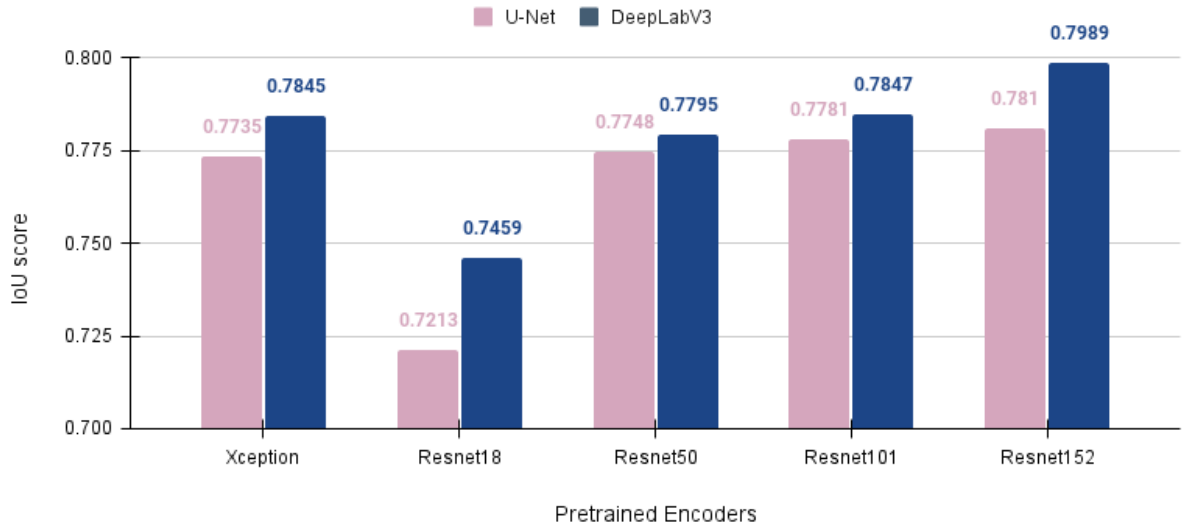


Fig. 9. Comparison of U-Net and DeepLabV3 decoders with following pretrained encoders: Xception, Resnet18, Resnet50, Resnet101, and Resnet152.

further improved using transfer learning.

Secondly, qualitatively comparison of the performance of the models are done. Fig. 8 illustrates the prediction of the two models on a sample image. It can be noticed that the baseline models perform reasonably well in capturing the majority of tree crowns, but their segmentation maps are not entirely precise due to the significant overlap between the surrounding vegetation and the tree crown. The presence of green-colored vegetation in the surrounding area poses a challenge for the models to accurately segment the tree crowns. However, this issue is effectively resolved by our top-performing model, which produces sharper edges and confines the map exclusively to the tree crowns.

4.4 Generalizability

In the domain of segmentation tasks, the term ”generalizability” pertains to a model's aptitude to precisely segment new, unobserved data that originates from a distinct dataset. In essence, a model that exhibits a high degree of generalizability is capable of performing accurately on an extensive range of data, not just the training data on which it was developed. The ability of a segmentation model to generalize is a crucial factor to consider when designing such models. If a model can only perform satisfactorily on the training data, its efficacy in real-world applications may be limited, particularly if it encounters data that significantly deviates from the training data. Hence, ensuring a model's ability to generalize to new data is essential to guarantee its practical utility.

To test the generalizability, our best segmentation model developed has been tested on another segmentation dataset that contains high-resolution satellite images of the Lower Saxony area of Germany. This dataset is different from the one used to train and validate the model, which means that it contains new and unseen data. To test the model on this new dataset, the images are first preprocessed and segmented using the trained model. The segmentation results are then compared to ground truth labels for the images, which provide the true segmentation masks. The result of testing our model on this new dataset's sample image is depicted in Fig. 11. The model's performance on this novel data is commendable, with an IoU score of 0.6871, demonstrating its ability to generalize well on unseen data. Although some areas in the lower right corner of the image are not segmented correctly, overall, the model performs optimally. Additionally, it could be seen that images in the unseen dataset has larger coverage of tree crowns (almost double) as compared to the dataset on which the model is trained on as shown in the Fig. 9 and Fig. 11.

4.5 Erroneous data

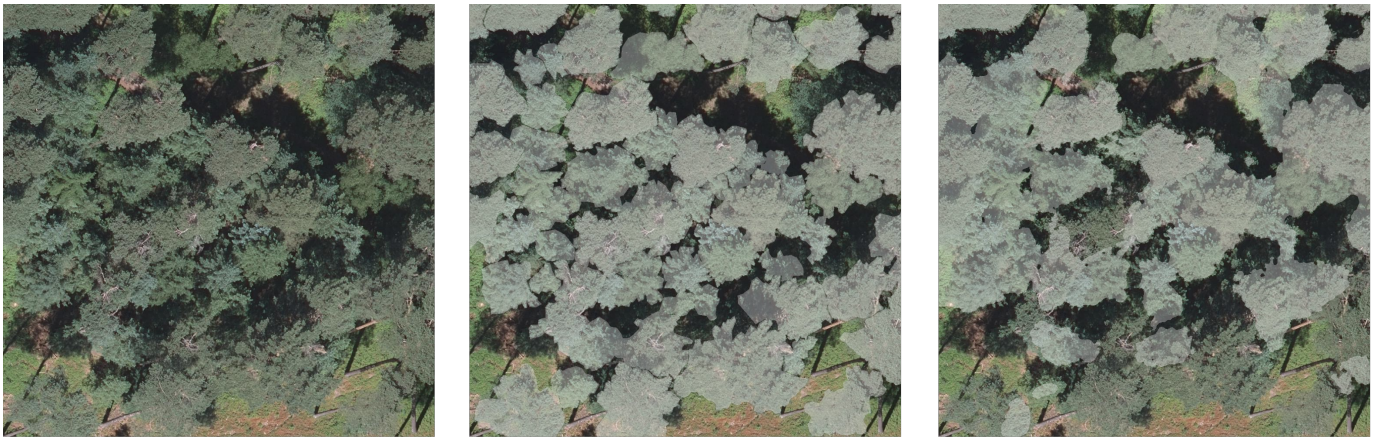
A key component for good model performance is the level of correctness of the labeled dataset [21]. manually annotated datasets are prone to human errors, and Fig. 10 illustrates that the mask in the ground truth has some questionable labels and suggests that the masks are partly erroneous. As it could be seen in the Fig. 10 center right, there is a questionable-elongated gap inside the tree group. Due to these errors, it is difficult to obtain very high accuracy in such datasets. Several methods have achieved very high performance in the segmentation tasks on satellite images. For instance, Sina et al. [40] reached tree cover extraction accuracies of 96-98 % combining a CNN with Object-Based Image Analysis. Martins et al. [29] reached an average accuracy of 90.63 % using a Dynamic Dilated Convolution Neural Network. However, given the small size and the problems in the labeled data, the accuracy of our best model of around 79.89 % can be considered good in comparison.

(a) (b)

Fig. 10. Example of potentially erroneous labels. (a) Original Image. (b) Mask with a questionable elongated gap inside of tree group.

5 Conclusion

This project presents a comprehensive study of experiments conducted with deep learning-based methods for tree segmentation in satellite images. This task is challenging due to the similarity in color between trees and their surroundings. To address this issue, the proposed methodology uses various encoder and decoder combinations based on ResNet, Xception, U-Net, and DeepLabV3, along with transfer learn-



Input Image **Ground Truth** **Predicted Map**

Fig. 11. Input image with ground truth and predicted label masked on input image taken from different dataset.

ing.

The study's results demonstrate that with small dataset size, transfer learning can significantly improve the accuracy of the segmentation task compared to models trained from scratch. Additionally, increasing the model's parameters often leads to improved segmentation maps. The study also highlights that different backbone architectures in the encoder and decoder play an essential role in improving the accuracy of the segmentation task. Also, the ability of a segmentation model to generalize is important for practical utility, and the best segmentation model was tested on a new dataset with commendable performance, indicating its high ability to generalize well on unseen data.

The study suggests that deep learning can effectively segment trees in satellite images for monitoring and managing forest resources, with potential for other remote sensing applications, and could contribute to understanding forest health and deforestation rates; future research can explore novel approaches to improve accuracy and advance the field.

6 Contribution

The project, which was a collaborative effort among several students from different countries, showcased each individual's skills.

- Pauline Drews, Master's degree in Ecosystem Analysis and Modelling
  Contributed on the ecological aspect to the project and studied on why deep learning is essential for forestry. Primarily focusing on the introduction and contextualization.
- Esther Cros, Master's degree in Bioinformatics
  Contributed in the methodology part for explaining the datasets studied and providing an overview of the frameworks while comparing them.
- Amit Bharti, Master's degree in Artificial Intelligence
  Worked on project pipeline, augmentation techniques and different hyperparameters to be used for segmentation task. In addition, trained the model from scratch using U-net architecture. Same things were translated into report in methods section.
- Rohit Yadav, Master's degree in Computer Science
  Used transfer learning to train and worked with different architecture to get the best results. Furthermore, evaluated the model to different datasets to valuate the model generalization capability. Wrote experiments, results and discussion section by comparing the various architectures studied.

The project's literature study, and conclusion, was a collaborative effort among all four individuals.

Acknowledgements

The authors wish to thank the organizers of the blended intensive course "Deep Learning in Forestry" that was carried out in cooperation with the Faculty of Forest Sciences and Forest Ecology (University of Göttingen), the Department of Biology (University of Bordeaux) and the Faculty of Science and Engineering (University of Groningen). Special thanks go to Nils Nölke, Matias Valdenegro-Toro, Jean-Christophe Taveau, and Max Freudenberg.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] M. Alberti and J. Marzluff. Ecological resilience in urban ecosystems: linking urban patterns to human and ecological functions. *Urban Ecosystems*, 7:241–265, 2004.

[3] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, B. C. Van Esesn, A. A. S. Awwal, and V. K. Asari. The history began from alexnet: A comprehensive survey on deep learning approaches. *arXiv preprint arXiv:1803.01164*, 2018.

[4] L. M. Anderson and H. K. Cordell. Influence of trees on residential property values in athens, georgia (usa): A survey based on actual sales prices. *Landscape and urban planning*, 15(1-2):153–164, 1988.

[5] E. Andersson, S. Barthel, S. Borgström, J. Colding, T. Elmqvist, C. Folke, and Å. Gren. Reconnecting cities to the biosphere: stewardship of green infrastructure and urban ecosystem services. *Ambio*, 43:445–453, 2014.

[6] E. Banzhaf and R. Hofer. Monitoring urban structure types as spatial indicators with cir aerial photographs for a more effective urban environmental management. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 1(2):129–138, 2008.

[7] T. Blaschke. Object based image analysis for remote sensing. *ISPRS journal of photogrammetry and remote sensing*, 65(1):2–16, 2010.

[8] M. A. Board. Millennium ecosystem assessment. *Washington, DC: New Island*, 13:520, 2005.

[9] B. Burkhard, J. Maes, M. Potschin-Young, F. Santos-Martín, D. Geneletti, P. Stoev, L. Kopperoinen, C. Adamescu, B. Adem Esmail, I. Arany, et al. Mapping and assessing ecosystem services in the eu-lessons learned from the esmeralda approach of integration. *One Ecosystem*, 3, 2018.

[10] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[11] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 11211 of *Lecture Notes in Computer Science*, pages 801–818. Springer, 2018.

[12] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[13] J. R. Clark, N. P. Matheny, G. Cross, and V. Wake. A model of urban forest sustainability. *Journal of arboriculture*, 23:17–30, 1997.

[14] J. F. Dwyer, H. W. Schroeder, and P. H. Gobster. The significance of urban trees and forests: toward a deeper understanding of values. *Journal of Arboriculture*, 17(10):276–284, 1991.

[15] F. J. Escobedo, T. Kroeger, and J. E. Wagner. Urban forests and pollution mitigation: Analyzing ecosystem services and disservices. *Environmental pollution*, 159(8-9):2078–2087, 2011.

[16] C. Glickman. Data augmentation in medical images, Nov 2020.

[17] E. Gómez-Baggethun and D. N. Barton. Classifying and valuing ecosystem services for urban planning. *Ecological economics*, 86:235–245, 2013.

[18] M. Havaei, A. Davy, D. Warde-Farley, A. Biard, A. Courville, Y. Bengio, C. Pal, P.-M. Jodoin, and H. Larochelle. Brain tumor segmentation with deep neural networks. *Medical image analysis*, 35:18–31, 2017.

[19] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2015.

[20] C. D. Ives, P. E. Lentini, C. G. Threlfall, K. Ikin, D. F. Shanahan, G. E. Garrard, S. A. Bekessy, R. A. Fuller, L. Mumaw, L. Rayner, et al. Cities are hotspots for threatened species. *Global Ecology and Biogeography*, 25(1):117–126, 2016.

[21] T. Jiang, M. Freudenberg, C. Kleinn, A. Ecker, and N. Nölke. The impacts of quality-oriented dataset labeling on tree cover segmentation using u-net: A case study in worldview-3 imagery. *Remote Sensing*, 15(6):1691, 2023.

[22] B. Jin, P. Ye, X. Zhang, W. Song, and S. Li. Object-oriented method combined with deep convolutional neural networks for land-use-type classification of remote sensing images. *Journal of the Indian Society of Remote Sensing*, 47:951–965, 2019.

[23] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[24] F. E. Kuo. Coping with poverty: Impacts of environment and attention in the inner city. *Environment and behavior*, 33(1):5–34, 2001.

[25] F. E. Kuo. The role of arboriculture in a healthy social ecology. *Journal of arboriculture*, 29(3):148–155, 2003.

[26] F. E. Kuo and W. C. Sullivan. Environment and crime in the inner city: Does vegetation reduce crime? *Environment and behavior*, 33(3):343–367, 2001.

[27] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[28] S. T. Lovell and J. R. Taylor. Supplying urban ecosystem services through multifunctional green infrastructure in the united states. *Landscape ecology*, 28:1447–1463, 2013.

[29] J. Martins, K. Nogueira, P. Zamboni, P. T. S. de Oliveira, W. N. Gonçalves, J. A. dos Santos, and J. Marcato. Segmentation of tree canopies in urban environments using dilated convolutional neural network. In *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, pages 6932–6935. IEEE, 2021.

[30] E. G. McPherson. Energy-saving potential of trees in chicago. Technical report, General Technical Report NE-186, 1994.

[31] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1520–1528, 2015.

[32] R. V. Platt and L. Rapoza. An evaluation of an object-oriented paradigm for land use/land cover classification. *The Professional Geographer*, 60(1):87–100, 2008.

[33] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 658–666, 2019.

[34] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

[35] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015*, volume 9351 of *Lecture Notes in Computer Science*, pages 234–241. Springer, 2015.

[36] S. Roy, J. Byrne, and C. Pickering. A systematic quantitative review of urban tree benefits, costs, and assessment methods across cities in different climatic zones. *Urban forestry & urban greening*, 11(4):351–363, 2012.

[37] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115:211–252, 2014.

[38] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):640–651, 2017.

[39] Statistisches Bundesamt (Destatis). Erläuterungen zum Indikator Anstieg der Siedlungs- und Verkehrsfläche“, 2023.

[40] S. Timilsina, J. Aryal, and J. B. Kirkpatrick. Mapping urban tree cover changes using object-based convolution neural network (ob-cnn). *Remote Sensing*, 12(18):3017, 2020.

[41] A. Tiwary, D. Sinnett, C. Peachey, Z. Chalabi, S. Vardoulakis, T. Fletcher, G. Leonardi, C. Grundy, A. Azapagic, and T. R. Hutchings. An integrated tool to assess the role of new planting in pm10 capture and the human health benefits: A case study in london. *Environmental pollution*, 157(10):2645–2653, 2009.

[42] K. Tubby and J. Webber. Pests and diseases threatening urban trees under a changing climate. *Forestry: An International Journal of Forest Research*, 83(4):451–459, 2010.

[43] M. G. Turner. Quantitative methods in landscape ecology: an introduction. *Quantitative methods in Landscape Ecology. The analysis and interpretation of landscape heterogeneity*, 1991.

[44] R. F. Young. Managing municipal green space for ecosystem services. *Urban forestry & urban greening*, 9(4):313–321, 2010.

[45] K. Yue, L. Yang, R. Li, W. Hu, F. Zhang, and W. Li. Treeunet: Adaptive tree convolutional neural networks for subdecimeter aerial image segmentation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 156:1–13, 2019.

[46] S. Zhang, Z. Ma, G. Zhang, T. Lei, R. Zhang, and Y. Cui. Semantic image segmentation with deep convolutional neural networks and quick shift. *Symmetry*, 12(3):427, 2020.

[47] S. Zhao, L. Da, Z. Tang, H. Fang, K. Song, and J. Fang. Ecological consequences of rapid urban expansion: Shanghai, china. *Frontiers in Ecology and the Environment*, 4(7):341–346, 2006.