A Deep Learning-based Approach for Tree Trunk Segmentation

Danilo Samuel Jodas*†, Sergio Brazolin†, Takashi Yojo†, Reinaldo Araujo de Lima†
Giuliana Del Nero Velasco†, Aline Ribeiro Machado†, João Paulo Papa*
*Department of Computing
São Paulo State University, Bauru-SP, Brazil 17033-360
Email: danilojodas@gmail.com, joao.papa@unesp.br
†Institute for Technological Research
University of São Paulo, São Paulo-SP, Brazil 05508-901
Email: {danilojodas,brazolin,yojos,reinaldol,velasco,asribeiro}@ipt.br

Abstract—**Recently, the real-time monitoring of the urban ecosystem has raised the attention of many municipal forestry management services. The proper maintenance of trees is seen as crucial to guarantee the quality and safety of the streetscape. However, the current analysis still involves the time-consuming fieldwork conducted for extracting the measurements of each part of the tree, including the angle and diameter of the trunk, to cite a few. Therefore, real-time monitoring is thoroughly necessary for the rapid identification of the constituent parts of the trees in images of the urban environment and the automatic estimation of their physical measures. This paper presents a method to segment the tree trunks in photographs of the municipal regions. To accomplish such a task, we introduce a semantic segmentation convolutional neural network architecture that incorporates a depthwise residual block to the well-known U-Net model to reduce the parameters required to create the network. Then, we perform a post-processing step to refine the segmented regions by removing the additional binary areas not related to the tree trunk. Lastly, the proposed method also extracts the central line of the identified region for future computation of the trunk measurements. Compared with the original U-Net architecture, the obtained results confirm the robustness of the proposed approaches, including similar evaluation metrics and the significant reduction of the network size.**

Index Terms—**Deep learning; convolutional neural networks; image processing; semantic segmentation; urban forest.**

I. Introduction

Smart city solutions are quickly gaining attention as part of several discussions towards a more sustainable urban ecosystem. The deployment of new emerging technologies is attracting the interest of many researchers and public administrators for smarter management of urban areas [1]. Among the new possibilities for promoting the rapid appraisal of the public areas, one can cite tree surveillance in urban streets. The assessment of urban trees is usually the principal concern of forest managers for the proper preservation of public spaces. Quality appraisal of the trees is often required to preserve the health of the citizens and avoid financial losses in case of any risk of falling.

Tree detection and urban forest surveillance are usually covered up in the state-of-the-art literature of remote sensing since aerial images are the most basic approach in the area [2]–[5]. However, images captured at the street level arise as a potential and inexpensive alternative in several tree monitoring tasks, including the mapping and inventory of the trees. The tree detection in images is usually performed considering the entire structure altogether without splitting their constituent parts to evaluate them separately. Such a process might facilitate the image segmentation and the measurements from solely the target part without a complex procedure to remove undesired elements in the scene. Examples of metrics acquired from specific parts of the tree include the Diameter at the Breast Height (DBH) and the tree angle, to name a few. The former consists of measuring the trunk width at 1.20 or 1.30 meters above the ground, while the latter includes the slope of the tree estimated from the trunk direction.

Computer-aided methods play an essential role to rapidly provide the most critical measurements for further analysis of the tree condition, including possible damages and the risk of falling [6]–[9]. Image processing and analysis, for instance, constitute one of the different options to identify trees in urban landscapes. However, automatic surveillance of the urban environment still poses a challenge because of the ecosystem heterogeneity of the city panorama. The presence of elements such as cars, light poles, and pedestrians, to cite a few, are the most challenging conditions to accurately identify trees in pictures, thereby representing a complex task even to the modern image processing methods.

The recent advances in deep learning techniques for object detection and segmentation in images have paved the way for more accurate results even in several circumstances that make the image analysis a complex task, including the low illumination conditions and the noisy artifacts. To such an extent, one can cite the advent of the Convolutional Neural Networks (CNN) as the well-known successful design in several image analysis domains, which comprises object classification and detection, and semantic segmentation as well. Regarding the latter, one can cite the U-Net [10] architecture as one of the most utilized CNN designed for object segmentation. Initially proposed for structure segmentation in medical images, the U-Net architecture gained attention in other applications, thereby confirming its robustness in several challenging conditions. Likewise, Badrinarayanan et al. [11] proposed a similar architecture named SegNet for semantic segmentation purposes. Unlike the U-Net model, the SegNet's decoder path only includes the indices of the max-pooling layer obtained by the encoder path. Consequently, this approach reduces the network size since the whole feature map is unnecessary for concatenation in the decoder path.

This paper presents a new approach for tree trunk segmentation using a deep CNN model based on the U-Net architecture composed of a residual depthwise convolutional block. The proposed model reduces the network size required in memory while providing effectiveness similar to the original U-Net architecture. Besides the trunk segmentation provided by the CNN model, this work also proposes refining the obtained results to remove other regions not belonging to the tree trunk. Furthermore, the proposed approach employs a further step regarding the central line extraction of the identified stem. This last step is conceived to avoid undesirable branches resulting from skeletonization methods, thus producing a central line that fits the trunk direction for future estimation of the slope toward streets or private and public properties.

The suggested architecture relies on the works of Gadosey et al. [12] and Pandey et al [13]. However, differently from the first study, our architecture incorporates a residual connection to avoid the gradient vanishing during the model's training. Also, we combine two extra convolutional operations to achieve more fine details from the input image.

The main contributions of this study are threefold:

- To propose a U-Net-based CNN model with reduced size to support the deployment in real-time applications;
- To propose a simple segmentation refinement with the connected component analysis; and
- To introduce a central line extraction approach that properly follows the trunk angle without branches in tip spots of the binary mask.

The paper is organized as follows: Section II presents the related works for tree detection in urban landscapes. Section III describes the proposed U-Net-based model and the steps employed to refine the segmentation results and extract the trunk's central line. Sections IV and V present the methodology and the results obtained from the performed experiments, respectively. Lastly, Section VI stresses the conclusions and future works.

II. Related works

As previously mentioned, a small number of studies have attempted to use digital photographs at the street-level view for tree detection and measurement of their physical aspects. Although the well-established analysis still relies on the Light Detection And Ranging (LiDAR) technology [14]–[18], street-level images are a cost-effective and non-expensive approach that gained attention owing to the recent advances in deep convolutional neural networks.

Teng et al. [19] proposed an image segmentation approach for identifying urban trees in ground-level images of the city landscape. The authors presented a skeletonization method to extract the tree trunk and detect the region above it as the tree canopy. Wang et al. [20] reported a similar approach that employs the L*a*b color space for identifying the image pixels related to the tree.

Branson et al. [21] proposed to use a Faster R-CNN model for identifying urban trees in images of the Google Street View (GSV) platform in combination with aerial photographs. Besides the tree detection, a Siamese CNN model was also employed for further assessing the differences in urban green landscapes over time. In a similar study, Laumer et al. [22] also performed tree detection in GSV images for the subsequent mapping of the tree geolocations with the corresponding street addresses previously recorded in earlier inventories.

Seiferling et al. [23] proposed to use computer vision methods for mapping the tree cover in street-level images. The proposed method initially performs the object segmentation by grouping pixels that share similar intensities at the super-pixel level. Super-pixels with similar features are then grouped into the same region for further determination of its class, which includes the tree itself. Afterward, tree cover estimation is performed using the ratio between the number of pixels classified as trees and the total number of pixels inside the whole image. Besides covering the tree inside one view of the streetscape only, the method also includes the appraisal of the neighbor images to handle the multiple tree occlusion and scale variations inside the scene.

Stubbings et al. [24] proposed identifying green areas at the street-level images through a deep CNN model based on a pyramid-parsing architecture called Pyramid Scene Parsing Network (PSPNet). This structure captures both local and global feature maps from images to further perform semantic segmentation. Comparing with two other methods used as a benchmark for green areas segmentation, the PSPNet model reported superior and accurate results to classify each image pixel as vegetation or non-vegetation.

Lumnitz et al. [25] proposed a CNN-based approach to detect and segment urban trees at the ground level of the street landscape. Using images provided by GSV and Mapillary platforms, the authors employed a Mask R-CNN for object detection and the subsequent instance segmentation for determining the geolocation of each identified tree. After applying the transfer learning of the Mask R-CNN with the weights obtained from the COCO dataset, the authors performed a second training with images of trees from the COCO Stuff dataset. Lastly, the model performs a fine-tunning procedure with street-level photographs of the GSV platform.

Despite the outstanding results reported in the mentioned studies, tree analysis still involves fine details that include identifying and evaluating other structures instead of the simple tree segmentation as a whole. Moreover, deep convolutional neural networks have gained attention either in detection or the pixel-wise classification for semantic segmentation of urban trees, which increases the ability for fine detail extraction and better handling in several landscape conditions.

III. Proposed approaches

This section describes the proposed pipeline that comprises the semantic segmentation and the post-processing step employed to refine the segmented objects and extract the central line of the trunk. Figure 1 shows the proposed pipeline for the tree trunk segmentation. The trunk region is manually cropped according to the dataset setup explained in Section IV-A.

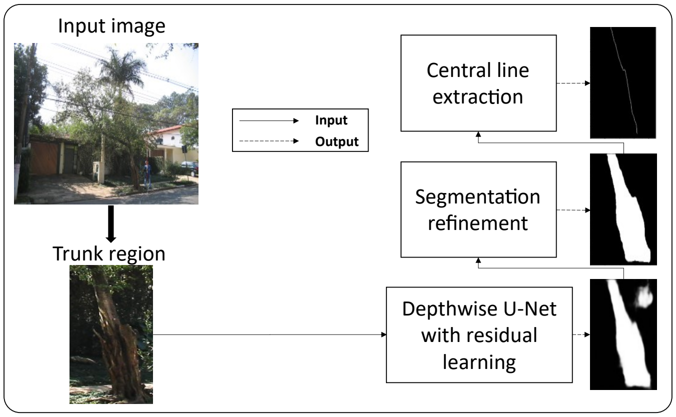


Fig. 1. Pipeline of the proposed approach.

A. *U-Net depthwise*

The U-Net model is a CNN architecture composed of encoding and decoding structures whose design delivers the feature extraction and the subsequent feature map reconstruction for object segmentation purposes. The encoder, or contracting path, is proposed for feature map extraction of the input image through a sequence of convolutions and max-pooling operations. Subsequently, the decoder takes the feature map as input for upsampling and further concatenating the restored map with the corresponding feature map obtained at the same level of the encoder path. This process produces a U-shaped architecture with symmetric layers connected level-by-level. Since it has been adopted in several studies for object segmentation purposes, the U-Net model was used as the base architecture for trunk segmentation in this work.

Compared to standard machine learning models, Convolutional Neural Networks provide robustness and remarkable results in several image analysis tasks. This singular capacity relies on the ability to extract high-level and fine-grained features from large image datasets. However, defining the number of layers in deep CNN architectures still resides a challenge considering the computational cost to perform the convolutional operations and issues related to deployment on devices with storage-limited capacity. The mentioned aspects involve the number of parameters required to build the model through multiple layers and standard convolutional kernels that lead to larger memory space to deploy the CNN model. Therefore, managing the computational complexity remains the foremost concern to reduce the architecture size.

The well-established approach to cope with the model parameter size consists of using the so-called depthwise convolution operation. Depthwise convolution is a class of convolutional operation wherein we perform the feature map extraction through a sequence of single convolutions in each color channel of the input image. Then, the convolved images are stacked together and taken as input for a point-wise convolution to produce a single feature map for one kernel. This approach aids in reducing the network parameters at the same time it saves memory space without decreasing effectiveness and prediction accuracy.

Since the U-Net architecture still relies on standard convolutions in each layer of the encoding and decoding paths, the network size becomes larger even for a relatively small number of layers. Therefore, we propose using depthwise convolutions on both sides of the model to reduce the architecture parameters. Figure 2 illustrates the suggested modification, while Figure 3 shows the convolutional blocks of the model.

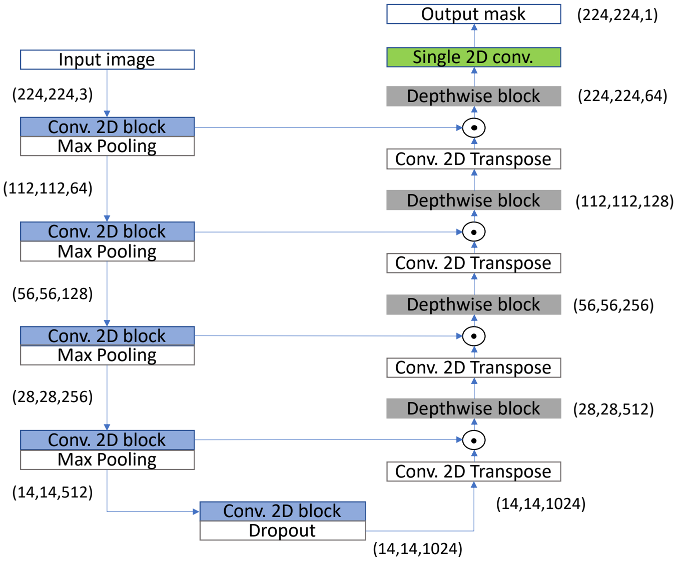


Fig. 2. U-Net-based architecture with depthwise convolutions and residual connections.

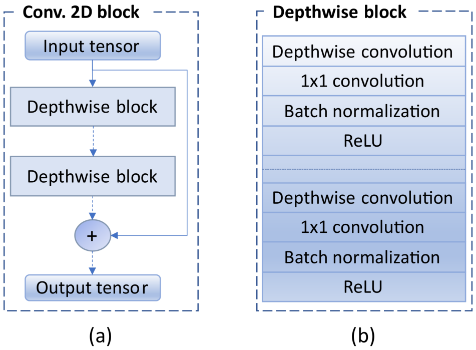


Fig. 3. Convolutional operations of the proposed architecture: a) Convolutional block with combined depthwise convolutions and residual connection; b) Block composed of double application of depthwise convolutions along with batch normalization and Rectified Linear Unit (ReLU) activation function.

As shown in Figure 2, the transposed convolutions are used to upsampling the feature map obtained at each level of the contracting path. Transpose convolutions are performed to reverse the downsampled feature maps attained from the encoder path. Since further details are lost during the reversal process, the decoder path also concatenates the upsampled feature map with the convolutions obtained from the encoder path.

The convolutional block (Figure 3a) is composed of two depthwise blocks for feature map extraction of the input tensor. The depthwise block (Figure 3b) includes two sequences of depthwise convolution, 1x1 convolution, batch normalization, and Rectified Linear Unit (ReLU) activation, thus matching the sequence adopted in the original convolutional block of the U-Net. As illustrated in Figure 2, the contracting path utilizes the convolutional block of Figure 3a, which leads to four depthwise operations to extract additional fine details from images. Also, we included a residual connection to eschew the gradient vanishing problem and increase the prediction effectiveness.

B. Segmentation refinement

Refining the segmentation results consists of removing additional binary objects adjacent to the identified trunk region. We used the well-known connected components analysis to find the different binary regions. In a nutshell, the method seeks the pixels whose intensities are similar and connected to each other. In binary images, the black color is often assigned to the background region, while the target objects receive the white color. Therefore, the connected component will find the white pixels within a n connectivity criterium, where n defines the neighbors of the white pixel under analysis. In this work, we consider an 8-connectivity pattern inside a 3x3 template centered at the pixel. Then, the pixels assigned to each connected region receive a unique label for helping the posterior searching. Afterward, we seek the region whose area is the largest among the others, which indicates the one related to the trunk. In the final step, the other binary regions are removed from the image. Figure 4 illustrates the results of the proposed refinement approach.

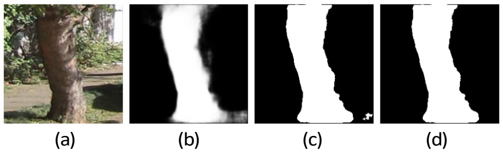


Fig. 4. Examples of further refinement: a) Input image; b) Semantic segmentation result; c) Threshold with value 0.5; d) Extra regions removed through the connected component analysis-based approach.

The first step consists of assigning 0 or 1 to each pixel of the image obtained from the U-Net depthwise CNN. This process uses the threshold value 0.5 to assign 1 if $I(x, y) \geq 0.5$, and 0 otherwise, where $I(x, y)$ stands for the pixel value at the (x, y) coordinate of the image I. Afterward, the connected component-based procedure seeks the region with the largest area and removes the small region located at the bottom right of the image (Figures 4c and 4d).

C. Central line extraction

In image processing and analysis, skeletonization is a process for extracting the medial axis of binary objects through a sequence of morphological operations that reduce the regions to a line while preserving connectivity between its points. Despite successfully used in several applications, the well-known skeletonization methods still suffer from branch formation at peak regions of the binary objects. Despite the recent progressions in the state-of-the-art works towards the refinement of the skeleton representation in binary images, border irregularities also pose a challenge since disturbances and noisy artifacts are still produced even in straightforward situations, thereby affecting the extraction of meaningful measures of the target objects. Therefore, we present a simple approach to extract the central line of the identified stem. In a nutshell, the method computes the average of the points located on both sides of the binary object's boundaries. Moreover, this process also fills the gaps in the line by using interpolation of the endpoints of the nearest disjoint segments, thus producing a smooth line positioned in the middle of the stem.

In the first step, we derive the right and left contours of the segmented binary mask of the trunk. Let $\mathcal{L}$ and $\mathcal{S}$ be the longest and the shortest contours, respectively, being $\mathcal{L} = \{l_1, l_2, ..., l_m\}$ and $\mathcal{S} = \{s_1, s_2, ..., s_n\}$ the sets of points of both boundaries of the mask. Then, the pair of points $s_j \in S$ and $l_i \in L$ are both used to determine the central point through a simple interpolation as follows:

$$c_i = l_i + (s_j - l_i) * \alpha \tag{1}$$

where c_i is the new point, $\alpha = 0.5$, $i = (1, 2, ..., m)$ and $j = (1, 2, ..., n)$, being m and n the size of $\mathcal{L}$ and $\mathcal{S}$, respectively. Figure 5 shows an example of the central line extraction.

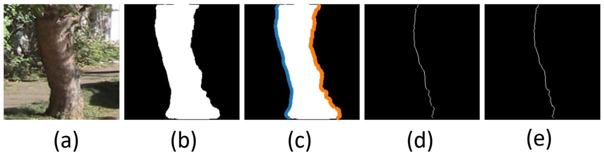


Fig. 5. Central line extraction from the trunk mask: a) Input image; b) Refined mask; c) Left (blue) and right (orange) contour points of the mask; d) Central line calculated from the left and right contours; e) Central line after filling the gaps.

IV. Methodology

This section presents the dataset and experimental setup adopted to perform the tests.

A. Dataset

Experiments were conducted over a dataset composed of patches of tree trunks extracted from images of São Paulo city, Brazil. The patches were manually delineated through a bounding box enclosing the entire trunk region, which comprises its base and the point where the canopy branches begin. The manual delineations were performed using the

LabelImg software[1]. This process resulted in 2,290 annotated trunk regions. Then, we cropped and saved the region inside the bounding box of each image for the subsequent manual annotation of binary masks that cover only the pixels of the tree trunk. In order to match the input size of the CNN models, all images were resized to a 224x224 resolution.

Due to the time-consuming process required for manual annotation, we selected only 801 cropped images for the trunk's binary mask generation, performed in the LabelMe software[2]. The first experiment involves the cross-validation procedure considering the split of the whole dataset into 5 folds for the initial model's assessment through different training and test sets configurations. Afterward, the entire dataset was shuffled and randomly split into training, validation, and test sets with a proportion of 70%, 15% and 15%, respectively, thereby leading to 561, 120, and 120 images into the respective divided groups for the final evaluation of the proposed refinement procedure.

B. Experimental setup

The proposed U-Net-based model has been implemented from scratch using Python 3.6 and Tensorflow Keras 2.3.0 without applying the transfer learning procedure. This new implementation has been necessary to support the depthwise convolutions and the lack of existing weights to fit the custom architecture. We performed all the tests in a Tesla© P4 GPU with 8 GB of RAM deployed on a computer equipped with an Intel© Xeon processor and 93 GB of RAM running the Ubuntu 16.04 Linux operational system.

This work compared the proposed architecture with the original U-Net model. Besides, the comparison also includes a straightforward U-Net-based design with one depthwise convolution block and no residual connections, thus following a similar architecture as the one proposed by Gadosey et al. [12].

For a reasonable comparison, we applied the same hyperparameters in all the experiments. The Adaptive Momentum Estimation (Adam) [26] has been employed to optimize the network's learning process with an initial learning rate of 0.0001 and a maximum of 1,000 epochs. The dropout rate of the proposed architecture is assigned to 0.3. Since the dataset is small, we also employed the data augmentation to each epoch of the training step. The image generator procedure includes the horizontal flip, brightness modification, Gaussian additive noise, Gaussian filter for smoothing, and histogram matching randomly applied considering a batch size of 4 images per training step at each epoch. We also used an early stopping criterium to avoid overfitting the model and stop the model's training procedure if no improvements are attained in the validation loss after 20 consecutive epochs. The training of the models involves the use of the binary cross-entropy as the loss function.

The experiments utilized the precision, recall, Dice Similarity Coefficient (DSC), and Intersection over Union (IoU) as validation metrics for evaluating the model effectiveness. All the metrics were calculated after binarizing each image at a threshold value of 0.5.

V. Experiments and results

Figure 6 depicts the segmentation results obtained from the proposed U-Net-based model. From Figure 6b, one can observe the well-behaved results in several landscapes and adverse conditions. The case shown in the third row, for instance, illustrates the difficulty regarding the presence of obstacles in front of the target tree trunk, thus posing a challenge for numerous applications. However, the proposed model performed properly even in such a challenging scenario, thus identifying the tree trunk correctly. Moreover, Figure 6c also shows the effectiveness of the proposed refinement step, which leads to binary masks that cover well the tree trunk in all illustrated examples (Figure 6d). Figure 6e also shows remarkable results obtained from the central line extraction method. One can notice the direction of the identified medial lines that fit well the trunk's slope in all the illustrated images.

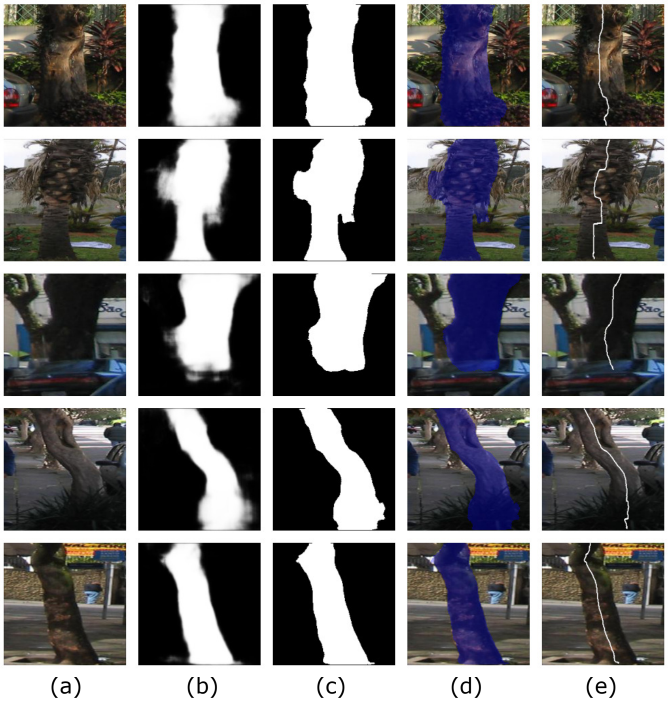


Fig. 6. Segmentation results obtained from the proposed Depthwise U-Net architecture with residual connection: a) Input images; b) Output obtained from the proposed model; c) Binary mask resulting from the refinement procedure; d) Binary mask overlayed on the input images; e) Central line (in white) extracted from the binary masks.

Figure 7 shows further examples obtained from the central line extraction approach. For comparison purposes, we also applied the well-known skeletonization method proposed in Zhang and Suen [27] to all masks shown in Figure 7b. Figure 7c shows results obtained from the central line extraction without applying the filling of disconnected regions. Considering the image shown in the first line, one can notice

[1] http://github.com/tzutalin/labelImg

[2] http://github.com/wkentaro/labelme

the recovered connections in the bottom part of the central line, thus confirming the importance of this step. Compared to the outcomes obtained by the skeletonization method, the proposed approach showed compatible results with the slope and the expected central position of the trunk. Furthermore, the central line extraction also avoided the inherent branches originated from the skeletonization method (Figure 7e).

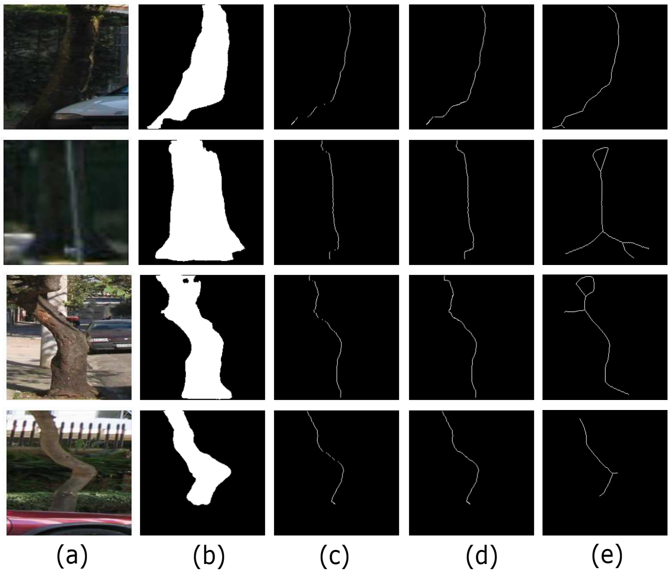


Fig. 7. Results obtained from the central line extraction algorithm: a) Input RGB images; b) Refined binary mask of the trunk; c) Central line with unconnected points; d) Gaps filled after interpolating the adjacent end points of the unconnected regions; e) Results from the skeletonization method proposed by Zhang and Suen [27].

To better understand the model's robustness, Table I presents the loss and the dice coefficient obtained from the test sets of each fold of the cross-validation procedure. These metrics involve the measurement considering a batch size of 32 samples at the model's evaluation step. One can notice the similar and even superior results of the proposed architecture against the two U-Net models in almost all folds. Furthermore, the U-Net-based models outperformed the results obtained from the SegNet architecture in all the evaluated folds considering the dice coefficient.

Table II also presents the evaluation metrics calculated from each architecture after applying the refinement procedure.

TABLE II
AVERAGE METRICS CALCULATED AFTER REFINING THE SEGMENTATION RESULTS OBTAINED FROM THE TEST SET.

	Precision	Recall	F1-Score	IoU
U-Net [10]	0.9368	0.9687	0.9476	0.8087
U-Net Depthwise [12]*	0.9282	0.9550	0.9351	0.8007
Ours	**0.9301**	**0.9673**	**0.9439**	**0.8147**
SegNet [11]	0.9453	0.9608	0.9480	0.8066

*We employed a similar architecture with transposed convolutions and batch normalization instead of group normalization.

It is noticeable the similar results obtained by all the architectures. In all cases, precision, recall and F1-Score achieved mean scores of more than 92%. Furthermore, Intersection over Union also showed promising results with more than 80% of overlapping with the manual delineations. One can notice the highest Intersection over Union obtained from the proposed model in comparison to the ones of the baseline architectures. Also, the proposed model showed similar performance compared to the original U-Net and the SegNet architectures. One can notice the slight difference considering precision, recall, and F1-score metrics.

Table III shows the number of parameters of each architecture and the average execution time considering the prediction for all images of the test set. The prediction process was performed tenfolds to evaluate the execution time variability for different runs.

TABLE III
NUMBER OF PARAMETERS AND THE AVERAGE TIME REQUIRED FOR THE BASELINE MODEL AND EACH DEPTHWISE ARCHITECTURE.

	# of parameters	Avg. time (in sec)
U-Net [10]	34,536,897	2.29±0.15
U-Net Depthwise [12]	9,517,919	2.24±0.38
Ours	12,403,679	2.95±0.33
SegNet [11]	29,458,949	2.38±0.72

Regarding the network parameters, one can notice the significant reduction in the size of the depthwise architectures compared to the baseline CNN models. U-Net Depthwise with Residual represents almost $\frac{1}{3}$ of the original U-Net's size and more than $\frac{1}{2}$ of the SegNet's size. The proposed architecture's size has a slight increase compared to the U-Net Depthwise model, which has a single convolutional block to perform the feature extraction from the input images. In contrast, the proposed model is composed of two blocks of depthwise convolutions, which leads to a small number of additional parameters. However, the rate of increase is nearly 1.3 for the proposed modification, which leads to improving the evaluation metrics shown in Table II. The execution time is almost similar for all architectures, considering the execution on the Tesla P4 GPU.

Figure 8 shows the loss, accuracy and DSC progression in the training step for each model. The proposed model starts to stabilize at 20 epochs of training with a slight increase in the validation loss. However, the early stopping criterium acts to avoid overfitting since no improvements were obtained after 40 epochs. Furthermore, accuracy and dice similarity also presents similar performance between the training and the validation curves. Compared to the other architectures (Figures 8a and 8b), one can observe that the depthwise residual model required fewer epochs to attain stabilization of the validation loss. While the original U-Net and the U-Net depthwise required, respectively, 60 and 50 epochs to stabilize, our model stopped at about 40 epochs with similar behavior considering the curves of the training progression. One can notice that the SegNet model required more than 150 epochs to stabilize and finish the training process.

Figure 9 shows the loss curves obtained from each U-Net-based architecture considering the 5-folds of the cross-

TABLE I

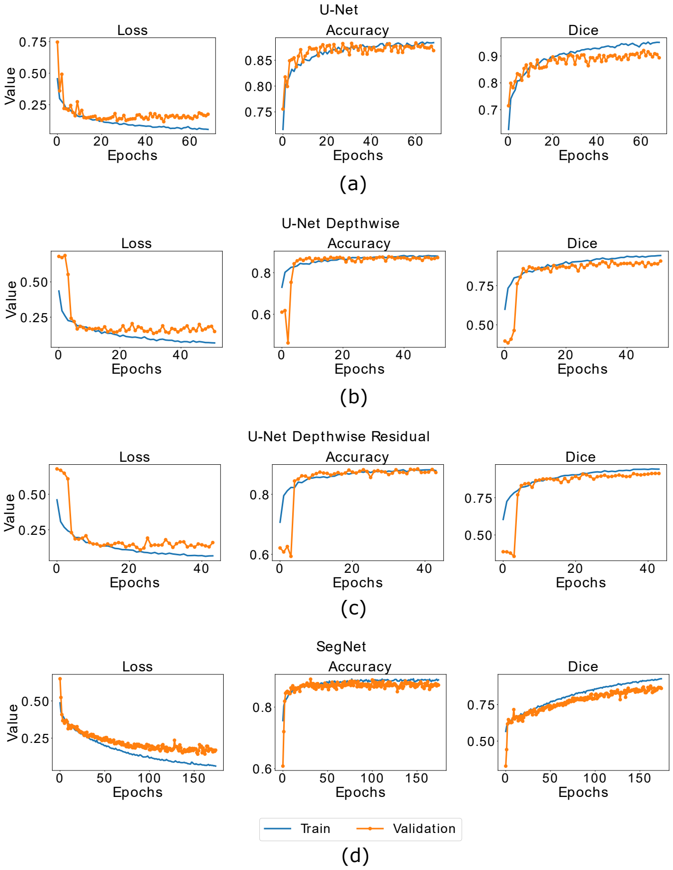
EVALUATION METRICS CALCULATED FROM EACH FOLD OF THE CROSS-VALIDATION.

	U-Net [10]		U-Net Depthwise [12]*		Ours		Segnet [11]	
	Loss	Dice	Loss	Dice	Loss	Dice	Loss	Dice
Fold 1	0.1538	0.9074	0.1841	0.8964	0.1943	0.9037	0.1778	0.8581
Fold 2	0.2261	0.8881	0.1739	0.8884	0.1581	0.8966	0.1913	0.8377
Fold 3	0.1910	0.8718	0.1931	0.8892	0.1673	0.9014	0.1863	0.8505
Fold 4	0.2285	0.8834	0.1684	0.8852	0.1891	0.8794	0.1786	0.8756
Fold 5	0.1551	0.9054	0.1743	0.8948	0.1437	0.9045	0.1737	0.8494
Average	**0.1909**	**0.8912**	**0.1788**	**0.8908**	**0.1705**	**0.8971**	**0.1815**	**0.8543**

*We employed a similar architecture with transposed convolutions and batch normalization instead of group normalization.

validation procedure. Despite the use of more epochs to complete the training of the first fold, the proposed architecture stabilized with less than 40 epochs in the other folds. In contrast, the standard and the depthwise without residual models required more epochs to stop the training. Furthermore, the curve behavior is similar across all architectures.

The obtained results emphasize the potential of the proposed approach for real-time applications with reduced network size and high precision in detecting the tree trunk in several conditions, thus promoting a relevant contribution to urban ecosystem management.



(a)

(b)

(c)

(d)

Fig. 8. Loss, accuracy and DSC training curves for each architecture: a) Original U-Net; b) U-Net with a single depthwise block; c) Proposed Depthwise U-Net with residual learning; d) SegNet.

VI. CONCLUSIONS AND FUTURE WORKS

Urban ecosystem surveillance continues an innovative topic of research in the context of intelligent city solutions. In this sense, this work presented an approach to support the automatic segmentation of the tree trunk in images of the urban environment using a U-Net-based model for semantic segmentation. We proposed reducing the network architecture through residual connections and depthwise convolutions that perform separated operations in each channel of the input image. Also, we employed a post-processing step to handle the additional segmented elements that do not belong to the trunk region. Lastly, the proposed approach also used a simple procedure to obtain the central line of the trunk for future estimation of its angle. This study only considered the tree trunk since this structure is initially diagnosed in the fieldwork operations to evaluate the tree condition.

Experiments conducted over few cropped images of the trunk region confirmed the effectiveness of the proposed approach considering the overlapping with the corresponding manual delineations and the similar results compared to the ones of the original U-Net architecture. Furthermore, the loss, accuracy, and Dice Coefficient curves of the tested models also showed the same behaviors along the epochs of the training step.

Future works will be conducted to automatically detect and crop the tree trunk region from the whole picture of the streetscape. The annotation of more images is also intended to increase the Intersection over Union and potentially offset the difference between the training and validation loss function. Besides, we plan to estimate the tree measurements for future assessment of the risk of falling, which plays an essential role in supporting the fieldwork team with several management tasks.

ACKNOWLEDGMENTS

The authors are grateful to FAPESP grants #2013/07375-0, #2014/12236-1, #2017/50343-2, #2019/18287-0 and #2019/07665-4 for supporting this research, as well as CNPq grants 307066/2017-7 and 427968/2018-6.

REFERENCES

[1] S. A. Nitoslawski, N. J. Galle, C. K. Van Den Bosch, and J. W. Steenberg, "Smarter ecosystems for smarter cities? A review of trends, technologies, and turning points for smart urban forestry," *Sustainable Cities and Society*, vol. 51, p. 101770, November 2019.

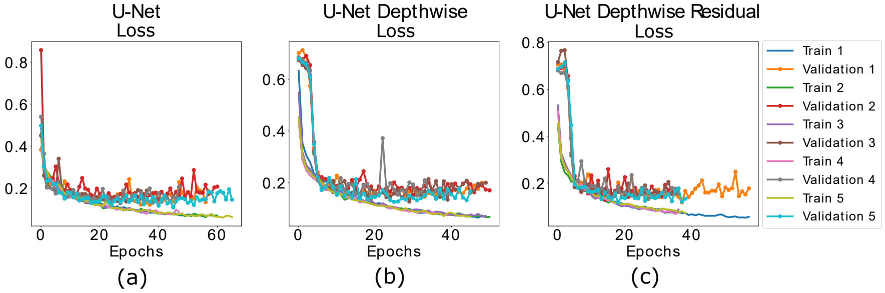


Fig. 9. Loss curves obtained from the cross-validation procedure for each U-Net-based architecture: a) Original U-Net; b) U-Net with a single depthwise block; c) Proposed Depthwise U-Net with residual learning.

[2] Y. Wang, J. Wang, S. Chang, L. Sun, L. An, Y. Chen, and J. Xu, "Classification of street tree species using uav tilt photogrammetry," *Remote Sensing*, vol. 13, no. 2, pp. 1–18, 2021.

[3] S. Briechle, P. Krzystek, and G. Vosselman, "Silvi-Net – A dual-CNN approach for combined classification of tree species and standing dead trees from remote sensing data," *International Journal of Applied Earth Observation and Geoinformation*, vol. 98, no. December 2020, p. 102292, 2021.

[4] K. Yarak, A. Witayangkurn, K. Kritiyutanont, C. Arunplod, and R. Shibasaki, "Oil palm tree detection and health classification on high-resolution imagery using deep learning," *Agriculture (Switzerland)*, vol. 11, no. 2, pp. 1–17, 2021.

[5] N. Guimarães, L. Pádua, P. Marques, N. Silva, E. Peres, and J. J. Sousa, "Forestry remote sensing from unmanned aerial vehicles: A review focusing on the data, processing and potentialities," *Remote Sensing*, vol. 12, no. 6, 2020.

[6] A. Dixit and Y. Nain Chi, "Classification and Recognition of Urban Tree Defects in a Small Dataset using Convolutional Neural Network, Resnet-50 Architecture, and Data Augmentation," *Journal of Forests*, vol. 8, no. 1, pp. 61–70, 2021.

[7] Y. Wei, H. Wang, K. F. Tsang, Y. Liu, C. K. Wu, H. Zhu, Y.-t. Chow, and F. H. Hung, "Proximity Environmental Feature Based Tree Health Assessment Scheme Using Internet of Things and Machine Learning Algorithm," *Sensors*, vol. 19, no. 14, p. 3115, July 2019.

[8] C. K. Wu, K. F. Tsang, Y. Liu, H. Wang, H. Zhu, C. H. Koo, W. H. Wan, and Y. Wei, "An IoT Tree Health Indexing Method Using Heterogeneous Neural Network," *IEEE Access*, vol. 7, pp. 66 176–66 184, 2019.

[9] H. Santoso, H. Tani, X. Wang, A. E. Prasetyo, and R. Sonobe, "Classifying the severity of basal stem rot disease in oil palm plantations using WorldView-3 imagery and machine learning algorithms," *International Journal of Remote Sensing*, vol. 40, no. 19, pp. 7624–7646, 2019.

[10] 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*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham: Springer International Publishing, 2015, pp. 234–241.

[11] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[12] P. K. Gadosey, Y. Li, E. A. Agyekum, T. Zhang, Z. Liu, P. T. Yamak, and F. Essaf, "SD-UNET: Stripping down U-net for segmentation of biomedical images on platforms with low computational budgets," *Diagnostics*, vol. 10, no. 2, 2020.

[13] R. K. Pandey, A. Vasan, and A. Ramakrishnan, "Segmentation of liver lesions with reduced complexity deep models," *arXiv preprint arXiv:1805.09233*, 2018.

[14] L. Terryn, K. Calders, M. Disney, N. Origo, Y. Malhi, G. Newnham, P. Raumonen, M. Å kerblom, and H. Verbeeck, "Tree species classification using structural features derived from terrestrial laser scanning," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 168, pp. 170–181, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0924271620302173

[15] W. Zhang, P. Wan, T. Wang, S. Cai, Y. Chen, X. Jin, and G. Yan, "A novel approach for the detection of standing tree stems from plot-level terrestrial laser scanning data," *Remote sensing*, vol. 11, no. 2, p. 211, 2019.

[16] J. Wu, W. Yao, and P. Polewski, "Mapping Individual Tree Species and Vitality along Urban Road Corridors with LiDAR and Imaging Sensors: Point Density versus View Perspective," *Remote Sensing*, vol. 10, no. 9, p. 1403, September 2018.

[17] C. Cabo, C. Ordóñez, C. A. López-Sánchez, and J. Armesto, "Automatic dendrometry: Tree detection, tree height and diameter estimation using terrestrial laser scanning," *International Journal of Applied Earth Observation and Geoinformation*, vol. 69, no. November 2017, pp. 164–174, 2018.

[18] L. Zhong, L. Cheng, H. Xu, Y. Wu, Y. Chen, and M. Li, "Segmentation of Individual Trees from TLS and MLS Data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 2, pp. 774–787, 2017.

[19] C.-H. Teng, Y.-S. Chen, and W.-H. Hsu, "Tree segmentation from an image." in *Proceedings of the 9th IAPR Conference on Machine Vision Applications, MVA 2005*, 2005, pp. 59–63.

[20] X.-s. Wang, X.-y. Huang, and H. Fu, "The study of color tree image segmentation," in *2009 Second International Workshop on Computer Science and Engineering*, vol. 2, 2009, pp. 303–307.

[21] S. Branson, J. D. Wegner, D. Hall, N. Lang, K. Schindler, and P. Perona, "From Google Maps to a fine-grained catalog of street trees," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 135, pp. 13–30, 2018.

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

[23] I. Seiferling, N. Naik, C. Ratti, and R. Proulx, "Green streets - Quantifying and mapping urban trees with street-level imagery and computer vision," *Landscape and Urban Planning*, vol. 165, pp. 93–101, September 2017.

[24] P. Stubbings, J. Peskett, F. Rowe, and D. Arribas-Bel, "A hierarchical Urban forest index using street-level imagery and deep learning," *Remote Sensing*, vol. 11, no. 12, pp. 1–22, 2019.

[25] S. Lumnitz, T. Devisscher, J. R. Mayaud, V. Radic, N. C. Coops, and V. C. Griess, "Mapping trees along urban street networks with deep learning and street-level imagery," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 175, pp. 144–157, September 2021.

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

[27] T. Y. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," *Communications of the ACM*, vol. 27, no. 3, pp. 236–239, 1984.