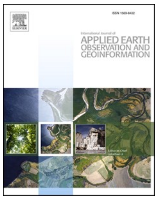
International Journal of Applied Earth Observations and Geoinformation 110 (2022) 102759

Contents lists available at ScienceDirect

International Journal of Applied Earth Observations and Geoinformation

journal homepage: www.elsevier.com/locate/jag



Line-based deep learning method for tree branch detection from digital images

Rodrigo Silva [a], José Marcato Junior [a], Laisa Almeida [a], Diogo Gonçalves [b], Pedro Zamboni [a], Vanessa Fernandes [c], Jonathan Silva [b], Edson Matsubara [b], Edson Batista [a], Lingfei Ma [d,*], Jonathan Li [e], Wesley Gonçalves [a,b]

[a] *Faculty of Engineering, Architecture, and Urbanism and Geography, Federal University of Mato Grosso do Sul, Av. Costa e Silva, Campo Grande, 79070-900 MS, Brazil*
[b] *Faculty of Computer Science, Federal University of Mato Grosso do Sul, Av. Costa e Silva, Campo Grande, 79070-900 MS, Brazil*
[c] *Faculty of Agricultural Sciences, Federal University of Grande Dourados, R. João Rosa Góes, Dourados, 79825-070 MS, Brazil*
[d] *Engineering Research Center of State Financial Security, Ministry of Education, Central University of Finance and Economics, Beijing 102206, China*
[e] *Department of Geography and Environmental Management, University of Waterloo, Waterloo, ON N2L 3G1, Canada*

ARTICLE INFO

Keywords:
Deep learning
Convolutional neural networks
Line detection
Tree branch

ABSTRACT

Preventive maintenance of power lines, including cutting and pruning of tree branches, is essential to avoid interruptions in the energy supply. Automatic methods can support this risky task and also reduce time-consuming. Here, we propose a method in which the orientation and the grasping positions of tree branches are estimated. The proposed method firstly predicts the straight line (representing the tree branch extension) based on a convolutional neural network (CNN). Secondly, a Hough transform is applied to estimate the direction and position of the line. Finally, we estimate the grip point as the pixel point with the highest probability of belonging to the line. We generated a dataset based on internet searches and annotated 1868 images considering challenging scenarios with different tree branch shapes, capture devices, and environmental conditions. Ten-fold cross-validation was adopted, considering 90% for training and 10% for testing. We also assessed the method under corruptions (gaussian and shot) with different severity levels. The experimental analysis showed the effectiveness of the proposed method reporting F1-score of 96.78%. Our method outperformed state-of-the-art Deep Hough Transform (DHT) and Fully Convolutional Line Parsing (F-Clip).

1. Introduction

Deep learning has been widely explored jointly with robotics in multiple areas, such in industry (Guan et al., 2021; Wang et al., 2021), agriculture (Yang et al., 2021;Chen et al., 2021;Lu et al., 2021;Tang et al., 2020;da Silva et al., 2019), and medicine (Shafiei et al., 2021; Luongo et al., 2021). Likewise, utility companies can benefit from these research areas. With the increase of cities, power grids have become larger and complex, making it difficult the manual maintenance of these systems. Robots with computer vision systems can be useful in grid maintenance.

Electric energy is one of the most critical assets in our daily lives. However, the electric power grid is very susceptible to situations like extreme weather events (Alam et al., 2020), leading to electricity shortages. Even though trees in cities are relevant, providing many advantages (Zamboni et al., 2021), tree branches can also become a problem when reaching the power lines. Mainly, electricity shortages are caused by tree branches that damage posts and wires. Thus, preventive maintenance is crucial to keep the work conditions of the grid and prevent possible accidents and shortages. The pruning of tree branches that get close to the power lines is part of the preventive maintenance, and it should be done frequently (Parent et al., 2019).

Nowadays, the cutting is done manually by a trained specialist. This manual method introduces risks for the professional responsible for the cut, is costly, hardware intensive, and time-consuming (Siebert et al., 2014). The maintenance can deal with different scenarios, depending on the state of the tree. In many cases, they will face limited vision due to overgrown branches and twigs, making their work even harder.

[*] Corresponding author.
E-mail addresses: rodrigo.a.silva@ufms.br (R. Silva), jose.marcato@ufms.br (J.M. Junior), laisa.fernanda@ufms.br (L. Almeida), diogo.goncalves@ufms.br (D. Gonçalves), pedro.zamboni@ufms.br (P. Zamboni), vanessafernandes@ufgd.edu.br (V. Fernandes), jonathan.andrade@ufms.br (J. Silva), edsontm@facom.ufms.br (E. Matsubara), edson.batista@ufms.br (E. Batista), l53ma@cufe.edu.cn (L. Ma), junli@uwaterloo.ca (J. Li), wesley.goncalves@ufms.br (W. Gonçalves).

https://doi.org/10.1016/j.jag.2022.102759
Received 20 December 2021; Received in revised form 23 March 2022; Accepted 24 March 2022
Available online 9 May 2022

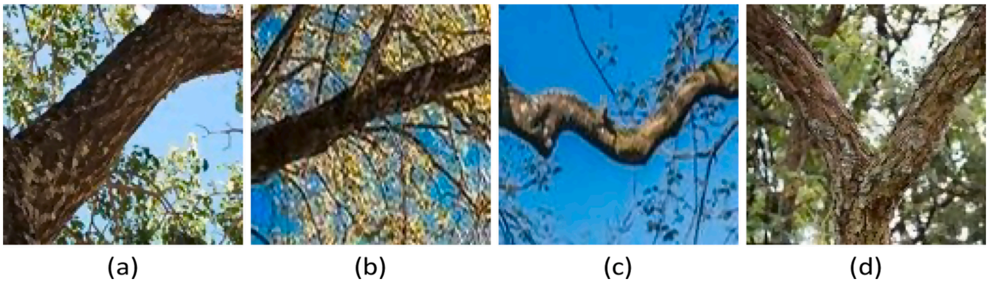


Fig. 1. Example of images in the dataset containing: (a) an isolated main tree branch, (b) a main tree branch with others in the background, (c) a curved, and (d) a bifurcated tree branch. These scenarios make it difficult to identify the tree branch orientation.

Depending on the branch's risk to the electricity network, the energy supply in the region is interrupted to properly clean the right of way, which ends up harming consumers. On the other hand, some norms establish the maximum time and amount of times that the energy concessionaire can be without serving its consumers during a specific period; this includes maintenance, accidents, and natural phenomena. If the company exceeds these limits, a fine is charged on top of the excess. In this context, methods that can automatically locate the tree branch without involving human beings are essential for safer and quicker power grid maintenance.

Deep learning approaches, mainly those based on convolutional neural network (CNN), have been successfully applied in remote sensing applications (Ma et al., 2019; Osco et al., 2021b; Yuan et al., 2021). When applying remote sensing to electrical engineering, previous works focused on detecting and classifying poles/towers in aerial images (Zhang et al., 2018b; Odo. et al., 2020; Gomes et al., 2020). Most of the studies for the tree trunk and branch detection were done for agricultural proposes (Zhang et al., 2018a; Majeed et al., 2018; Amatya et al., 2016; Zhang et al., 2021; Majeed et al., 2020) and are mainly based on semantic segmentation techniques (Zahid et al., 2021). A disadvantage of semantic segmentation techniques is the requirement of dense labeling, i.e., each pixel on images has to be labeled. Also, these studies often work only with a single tree species, limiting the applications for more heterogeneous situations. For automatic pruning, some studies reconstructed a 3D model of the trunks and branches in precision farming (Karkee and Adhikari, 2015; Karkee et al., 2014). Despite these initial efforts in agriculture, there is a lack of studies for detecting tree branches with the purpose of power grid maintenance.

Few studies aim to automate the pruning of trees in power grids context. In Molina and Hirai (2017) a drone-based mechanism was developed to fly to the branch of interest and grab it to perform the cut; however, the entire process is manual, it needs a person to fly it to the tree branch, grab it and cut it. In Straub and David Reiser (2021,) a LiDAR (light detection and ranging) is used to create a 3D map of the tree and then decide which branch to cut, which is very interesting, but they need to teach the system how to position the saw and make it autonomous. Also, LiDAR sensors are expensive, and deep learning-based methods applied to 3D point cloud segmentation are less mature when compared to image-based solutions (Guo et al., 2020). In order to automate the pruning process, it is essential to correctly detect the tree branch and the point where we need to cut the branch. To that end, RGB sensors present themself as cheaper and easier to process than sensors like LiDAR.

The present proposal is a step towards automatic branch pruning. This scenario considers a robotic arm to cut the branches automatically. Here, our focus is on the vision sensor, which is one of the components of a robotic pruner (Zahid et al., 2021). A robotic arm with a camera attached is positioned by an operator pointing near the branch of interest. Due to laws and environmental issues, only the branches at risk with the energy grid should be pruned. Currently, a technician handles the robotic arm that grabs the branch to be pruned. However, in many cases, the technician has the field of vision obscured while operating the arm from the ground. With the image from the camera (robotic arm), the system needs to determine the cut point and angle supporting the fine adjustment of the arm. With these information, the robotic arm automatically prunes the branch. Therefore, this study focuses on the following problem: giving an image from the camera attached to the tip of the robotic arm, which is the branch orientation, and where is the grip position to perform the cut? Here, we propose a novel method capable of detecting a tree branch and consequently its orientation and the cut point. The proposed model combines CNN with Hough Transform and a grasping point propose module. Thus, our model can be used by utility companies to increase safety and efficiency in the tree pruning process and grid maintenance by automating the process and reducing the need for human intervention.

To the best of our knowledge, no prior study has focused on proposing a deep learning method for supporting the pruning of tree branches based on RGB images in power grids context. We propose a deep learning line-based extraction method, which is few explored in the computer vision community compared to other tasks. Unlike previous studies (see review articles (He and Schupp, 2018; Zahid et al., 2021) for apple trees), we focus on detecting tree branches in different tree species, which is a more challenging task. We evaluated our model in different scenarios, presenting a quantitative and qualitative analysis, and compared it with the state-of-the-art method for line detection. Finally, a single line as ground truth facilitates the labeling of images (two points) and presents an interesting scientific challenge (weakly supervised learning). In addition, our main objective is to detect the angulation of the tree branch to supply the robotic arm, which can be accomplished with a single line. In fact, the branch's dense labeling (semantic segmentation), commonly used in this context (Zahid et al., 2021), can make it easier to detect. However, the results showed that, for our application, the use of a single line is sufficient. Thus, it would not be necessary to predict all the pixels of the branch (optimal variance related to the thickness), but only the central pixels.

In summary, our original contributions are described as follows:

1. Construction and labeling of an image dataset to detect tree branches;
2. Development of a new lightweight method based on deep learning for straight line detection;
3. Results superior to the state-of-the-art in detecting tree branches using straight lines.

2. Materials and Methods

2.1. Image Dataset

To achieve our goal, we build a novel dataset of tree branch images[1]. As far as we know, this is the first dataset of images for this task. For this

[1] https://sites.google.com/view/geomatics-and-computer-vision/home/datasets

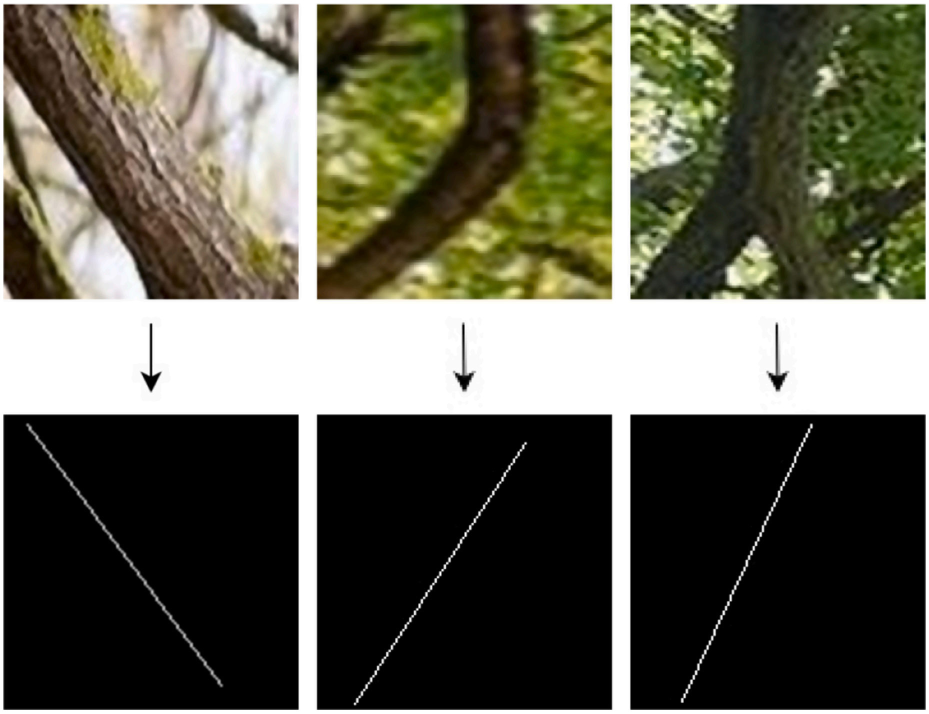


Fig. 2. Examples of labeling a straight line for the main tree branch.

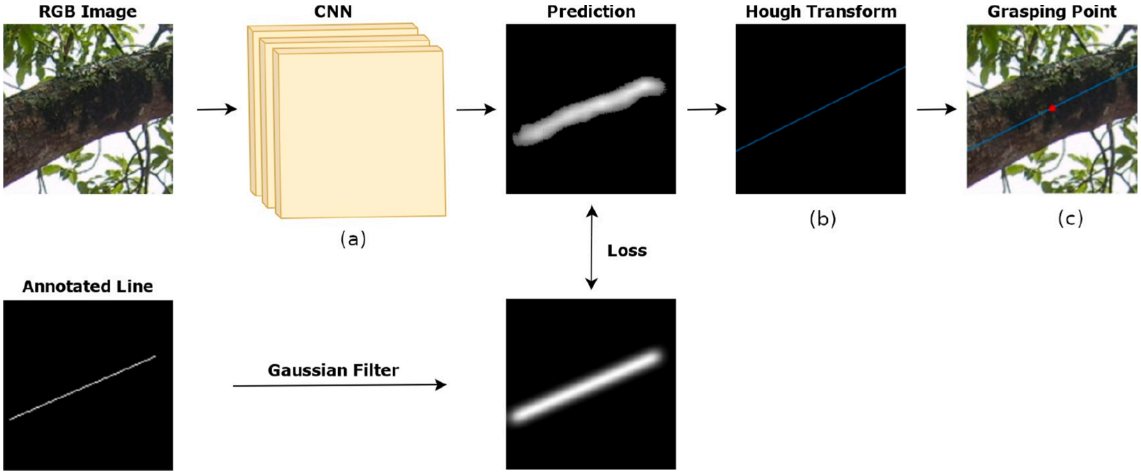


Fig. 3. Proposed method to estimate the line of the main tree branch in an image. The (a) first step performs the pixel-by-pixel prediction using a CNN, while the (b) second step estimates a straight line using the Hough transform. In the (c) last step, a point belonging to the line is calculated for the branch grip.

purpose, we crawled images using related phrases and words as search. Afterward, a data cleaning step discarded repeated or out-of-context images. The images were resized to 128 × 128 pixels.

The resulting dataset is composed of 1868 images, where each image contains at least one main tree branch. As the objective is to identify the main tree branch, the remaining must be ignored by the methods. Fig. 1 shows four image examples of the dataset, the first (Fig. 1a) with a single tree branch in the foreground, and the second (Fig. 1b) with many secondary tree branches close to the primary. The Figs. 1c and 1d illustrate some challenging images that include different shapes of branches. Also, there are images with variations in lighting conditions, weather conditions (e.g., snow, sun), capture devices, image quality, blurring, and different backgrounds.

All images were manually labeled with the direction of the main tree branch using a straight line. For forked or curved branches, the longest apparent section of the branch has been labeled with a straight line. Fig. 2 shows the labeling for dataset images. Labeling with a straight line is important to define the grip angle but also to simplify the labeling process. On the other hand, simplified labeling poses a challenge to the learning of automatic methods.

2.2. *Proposed Method*

The proposed method can be described in three main steps as shown in Fig. 3. The first step is to predict the probability that a pixel belongs to the line representing the main tree branch in the image. Due to recent results, this prediction is performed by a Convolutional Neural Network (CNN) as shown in Fig. 3(a). In the second step, the Hough transform is applied in the prediction to find a line (Fig. 3(b)). Finally, a point in the line is chosen for the grip (Fig. 3(c)). The sections below describe each step in detail.

The labeled line of the main tree branch is one pixel thick (Fig. 2). As

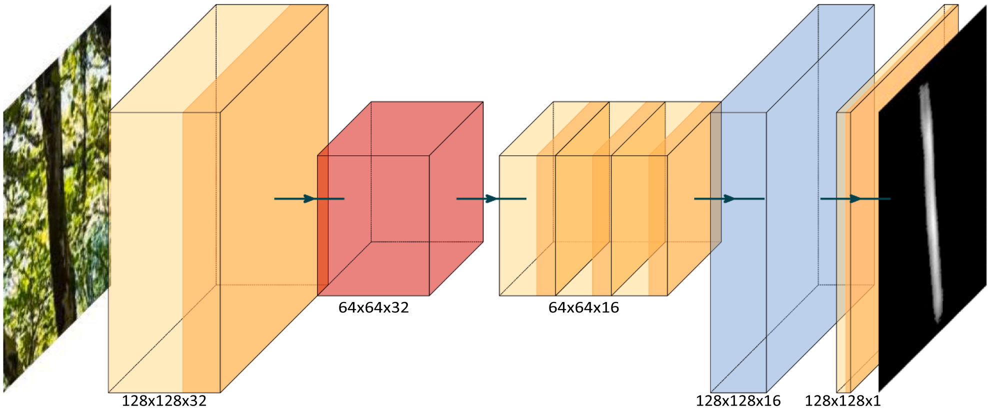


Fig. 4. CNN architecture for pixel-wise line prediction. It is composed of eight layers: five convolution (orange), one max-pooling (red) and one transposed convolution.

the imbalance between pixels belonging to the line and the background is large, CNNs have difficulty learning such scenarios. The CNN can simply predict all pixels as background and still get a low loss, causing learning not to occur correctly.

To solve this issue, we dilate the line by applying a Gaussian kernel with standard deviation σ to each pixel belonging to the line. In ground truth, the central part of the line has higher values that decrease as it moves away from the labeled line (this drop is controlled by the standard deviation σ). Due to this drooping behavior of the Gaussian kernel, it is not necessary to cover the entire branch in the ground truth, but only a region larger than the one-pixel-thick line. Therefore, the σ can be empirically determined in the validation set without a great influence on the results.

2.2.1. *Straight line prediction*

This step consists of predicting a line that describes the main tree branch in the image. For this, a CNN was proposed for pixel-wise prediction. Each value of the feature map represents the probability of the pixel belonging to the line. The CNN is composed of the following layers as shown in Fig. 4. The input image is initially processed by a convolution layer with 32 filters of size 3×3, and its dimension is halved (64×64) with a max-pooling layer. Then, the feature map goes through three convolution layers with 16 filters each for extracting deep features. A transposed convolution increased the resolution of the feature maps returning to the original size of 128×128 pixels. Finally, the last convolution layer containing one filter is applied to predict the probability that each pixel belongs to the tree branch line. All convolution layers use the PReLU activation function except the last layer that uses the ReLU activation function. We evaluated the CNN with ReLU, PReLU, and other activation functions, but PReLU provided the best results in the validation set. Therefore, CNN prediction $P(x,y)$ corresponds to the probability that pixel x,y belongs to the line representing the main tree branch.

The CNN proposed above is a lightweight version that can be easily embedded on devices. There is also the possibility to use heavier CNNs at this stage. To evaluate this possibility, we also used UNet (Ronneberger et al., 2015) with EfficientNetB0 as the backbone for line prediction.

2.2.2. *Hough Transform*

The CNN prediction does not directly represent a straight line or tree branch orientation despite obtaining good results. Thus, the second step of the proposed method applies the Hough transform to estimate a line from the prediction $P(x,y)$ for each pixel.

Before applying the Hough transform, we segment the prediction P through a threshold τ. All pixels with probability $P(x,y)$ greater than or equal to τ are maintained according to Eq. 1. To improve segmentation quality, we propose to use a variable threshold automatically calculated for each image. Instead of using a fixed threshold, a variable threshold is calculated as $\tau = 0.35 \cdot \max_{x,y} P(x,y)$. Thus, the binarized prediction maintains only the regions with a high probability of occurrence. The variable threshold directly affects the quality of the straight line generated by the Hough transform.

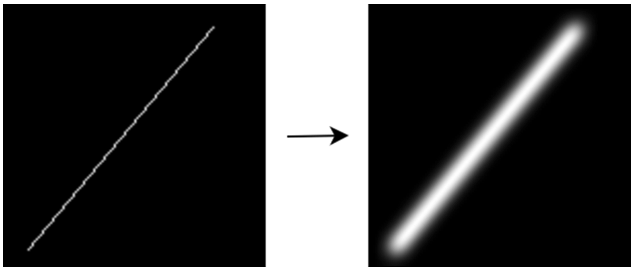


Fig. 5. Result of applying the Gaussian to the labeled line ($\sigma = 4.0$).

$$Q(x,y) = \begin{cases} 1, & \text{if } P(x,y) \geqslant \tau, \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

where $P(x,y)$ is the probability of the pixel x,y belonging to the tree branch line and Q the segmented prediction keeping the pixels with higher probability.

Given the binary prediction Q, the Hough transform is applied to estimate the most probable line. In this transform, a line represents the size of the segment r and the angle θ with the x-axis. Thus, the representation of the line is given by $r = x\cos\theta + y\sin\theta$. Pixels in Q vote for candidate lines represented by r and θ as per the previous equation. The most voted r,θ pair correspond to the most likely lines in the image. As our problem is identifying the line of the main tree branch, we only get the most likely line L.

2.2.3. *Grasping*

The previous steps of the proposed method provide the most likely tree branch line. This line provides the direction and position of the tree branch for positioning and angling both the grapple and the saw to perform the branch cut.

In addition, we propose a point on the tree branch for grasping. For this, consider the pixels (x,y) belonging to the most likely line L, and the prediction P performed by the CNN. The point $(x,y) \in L$ with the highest probability on P is defined as the grip point p_g, i.e., $p_g = \text{argmax}_{(x,y) \in L} P(x,y)$. Fig. 3(c) illustrates the grip point in red. As the point p_g has the highest probability in the CNN prediction, this point is the most likely to belong to the branch and can be considered a good location for grasping.

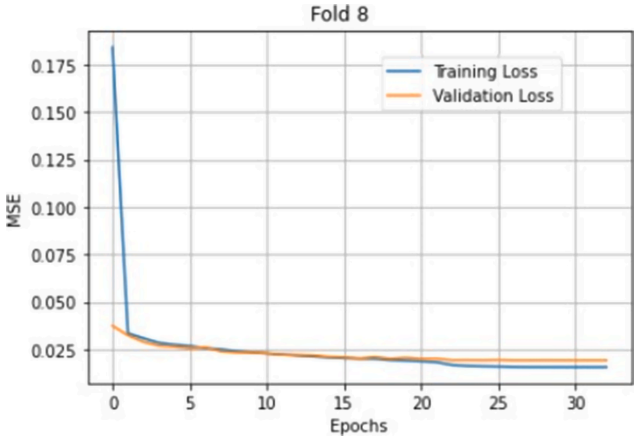


Fig. 6. Loss function during training in a fold of the cross validation. The loss rapidly decreased and stabilized indicating that the training process were adequate.

2.3. Experimental Setup

2.3.1. Proposed Method Training

To solve the imbalance between pixels belonging to the line and the background, we dilate the line by applying a Gaussian kernel with standard deviation σ to each pixel belonging to the line. Experiments showed that the CNN has better convergence for $\sigma = 4.0$. In Fig. 5 we have the annotation of a line and the result after applying the Gaussian kernel. It is possible to observe that the line has a greater thickness and the values are not constant, which also helps the CNN learn. The central part of the line has higher values that decrease as it moves away from the original line. We also applied data augmentation by rotating the images at 90, 180, and 270-degree angles, thus increasing the dataset by a factor of 4 to extend the training set.

The training procedure used MSE as loss function (*Mean Squared Error*), Adaptive Gradient Clipping (Brock et al., 2021) and the Adam optimizer with an initial learning rate of 10^{-3} with a reduction by a factor of 10 on a plateau measured in the loss of the validation set, and a batch size of 128. We used Early Stopping (10 epoch patience), which continuously operated before epoch 50 due to the fast convergence with the ground truth improved using line expansion.

2.3.2. Evaluation Metrics

To evaluate the performance of the proposed method, we use the k-fold cross-validation with $k = 10$. Thus, the results are the average of the folds. As evaluation metrics, we compare the predicted and labeled line through Precision (P), Recall (R) and $F1$-score according to Eqs. (2)–(4), respectively.

$$P = \frac{TP}{TP + FP} \tag{2}$$

$$R = \frac{TP}{TP + FN} \tag{3}$$

$$F1 - \text{score} = \frac{2TP}{2TP + FP + FN} \tag{4}$$

where TP, FP and FN correspond to True Positives, False Positives and False Negatives, respectively.

To calculate TP, FP and FN for straight lines, we follow the methodology suggested in (Osco et al., 2021a). A true positive TP represents a point on the predicted line at a maximum distance δ from any point on the labeled line. A false positive FP represents a point on the predicted line that is not close (i.e., distance greater than δ) to any point on the labeled line. Finally, a false negative FN represents a point on the labeled line that is not close to any point on the predicted line. The value δ indicates the maximum acceptable deviation between the predicted and labeled line points. During the experiments, we set the metrics as $\delta = 10$, 15 and 20 pixels.

Table 1
Result of the proposed method for each fold of the cross validation using the angular precision (A), precision (P), recall (R) and F1-score with $\delta = 15$.

Fold	A	P	R	F1-score
Fold 1	0.8496	0.9173	0.8930	0.8920
Fold 2	0.8625	0.9201	0.8966	0.8974
Fold 3	0.8465	0.9085	0.8960	0.8941
Fold 4	0.8420	0.9142	0.8913	0.8882
Fold 5	0.8420	0.9033	0.8756	0.8767
Fold 6	0.8521	0.9127	0.8994	0.8980
Fold 7	0.8317	0.9069	0.8942	0.8886
Fold 8	0.8587	0.9268	0.9091	0.9079
Fold 9	0.8407	0.9147	0.8879	0.8897
Fold 10	0.8509	0.9338	0.9013	0.9030
$\mu(\pm\sigma)$	0.84($\pm$0.009)	0.91($\pm$0.009)	0.89($\pm$0.009)	0.89($\pm$0.009)

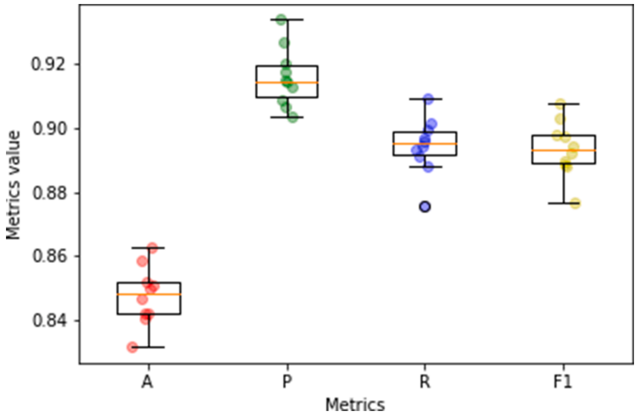


Fig. 7. Results for each ten-fold cross-validation (represented as dots) considering the Angular Precision (A), Precision (P), Recall (R) and F1-score metrics.

Since the angle of the proposed line is important for the correct grip of the tree branch, we use the Angular Precision (also used in Zhao et al. (2021)) that compares the angular difference between the predicted and labeled line (Eq. 5). We can see that this metric normalizes values between 0 and 1.

$$A = 1 - \frac{|\theta_g - \theta_h|}{\pi/2} \tag{5}$$

where θ_g and θ_h correspond to the angle of the labeled and predicted line, respectively.

3. Results and Discussion

During the experiments, the division of the training and testing set followed cross-validation with ten folds. Fig. 6 presents the loss function obtained during the training of the proposed method in folds eight. The curve behavior is adequate as the loss drops and stabilizes during the training epochs. Furthermore, we can observe that the loss in the training and validation set is close, showing the generalization of learning without overfitting. The other folds had similar behavior.

3.1. Quantitative Analysis

To quantitatively assess the generalization of the proposed method. Table 1 and Fig. 7 presents the results for each fold of the cross-validation. We can observe that the proposed method achieved strong results even for a considerable variation in the test set. Considering the average results, the angular accuracy is 0.84, and the F1-score is 0.89

Table 2
Average of F1-score, Recall and Precision for each threshold (δ).

Metrics	$\delta = 10$	$\delta = 15$	$\delta = 20$
F1-score	0.8197	0.8936	**0.9257**
R	0.8192	0.8945	**0.9268**
P	0.8516	0.9158	**0.9430**

with small standard deviation values showing consistent results among different folds. These results demonstrate that the tree branch and its orientation are well detected by the proposed method.

Previous results were obtained using $\delta = 15$; therefore, the metrics considered a maximum deviation of up to 15 pixels. Table 2 presents the average of the F1-score, Recall and Precision obtained for the cross validation folds using $\delta = 10, 15, 20$. Therefore, it is possible to observe the variation of the results considering more adjusted lines ($\delta = 10$) or more distant ($\delta = 20$) concerning the ground truth. The proposed method presents relevant results when considering more adjusted lines, with F1-score, R and P of 0.8197, 0.8192 and 0.8516.

3.2. Qualitative Analysis

The proposed method was tested qualitatively on an entirely new image set containing 36 images of tree branches under uncontrolled capture conditions. The metrics in this set achieved good values with F1-score of 0.9678, R of 0.9674 and P of 0.9698, considering $\delta = 15$.

Fig. 8 shows four examples of this test. The first column shows the input image where the prediction with the proposed method was performed. In the second and third columns, we can see the prediction and line generated by the Hough transform together with a point for the grip. Finally, we show the ground truth line (orange) and the predicted line (blue) in the last column. These results indicate that the proposed method successfully generalizes the learned features, despite being relatively simple and efficient.

The grip point was also suitable for being inside the tree branch (red dot in the third column of Fig. 8). Since this point computes the highest

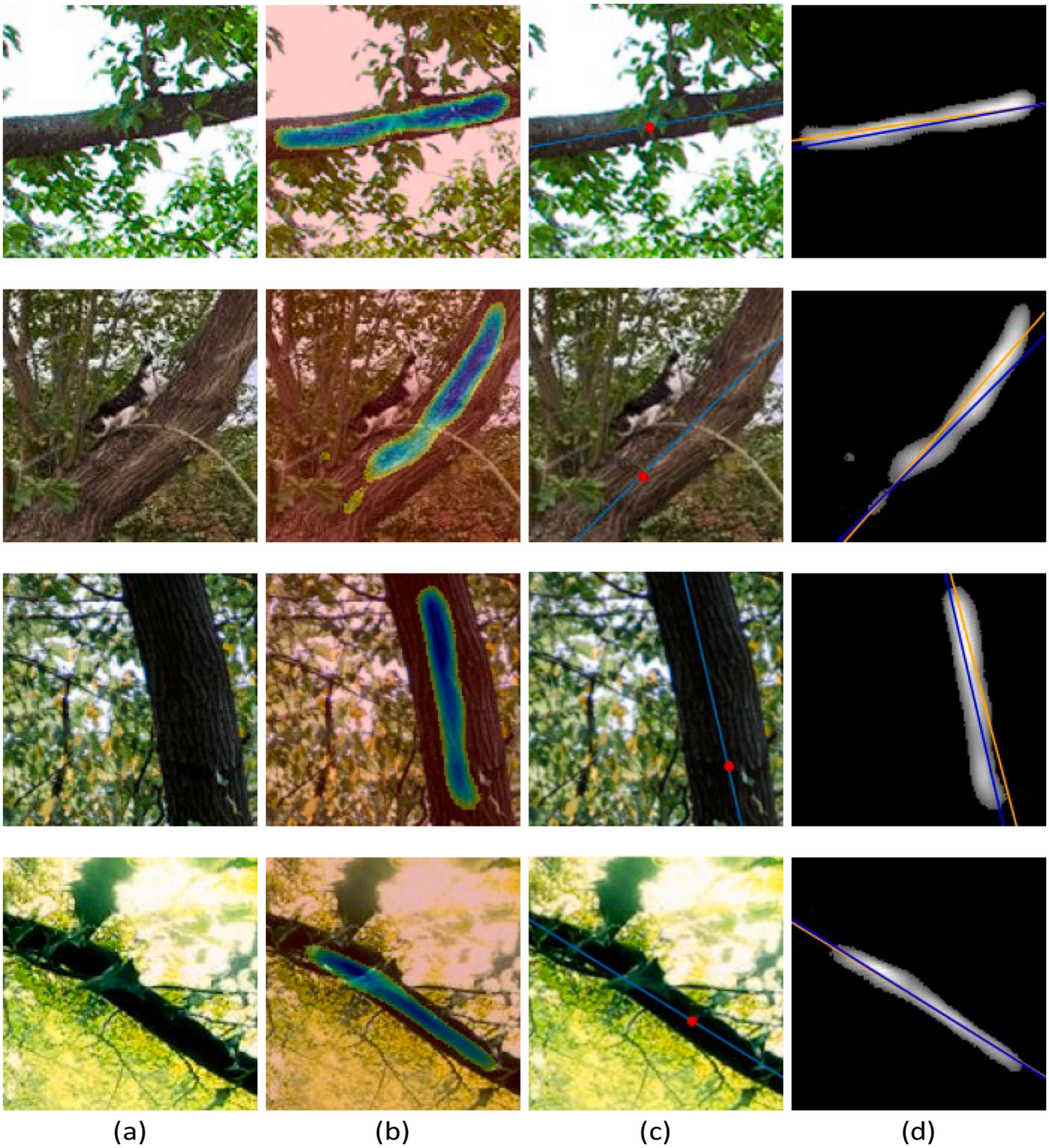
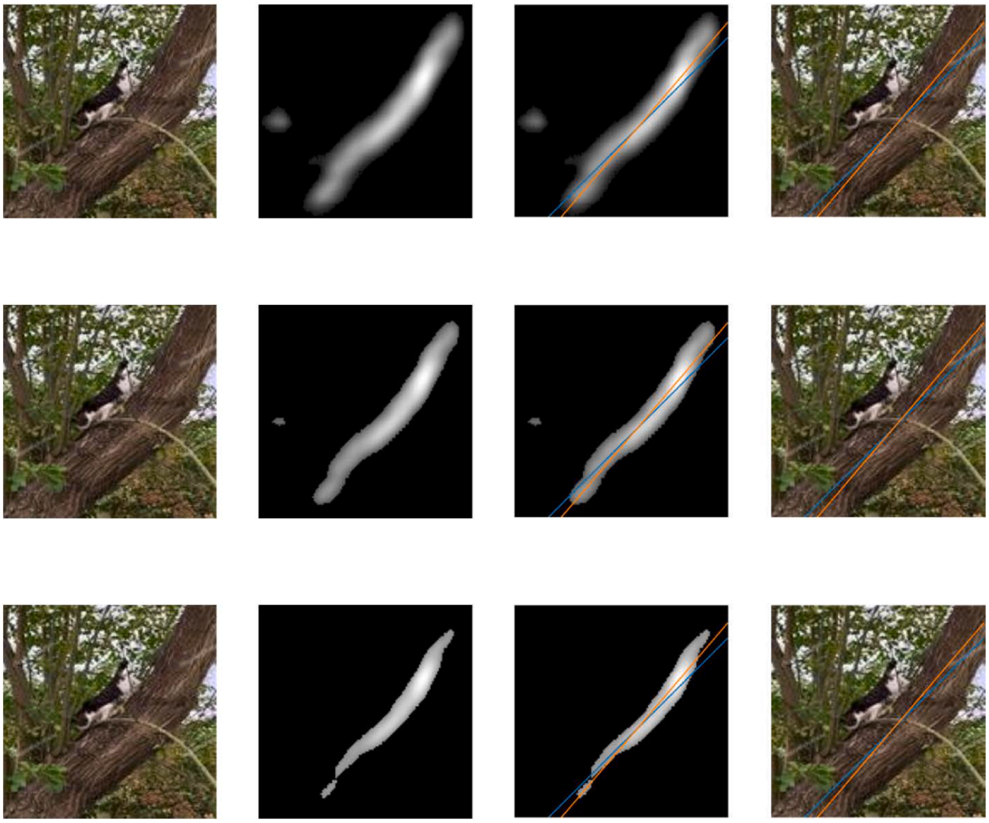


Fig. 8. Examples of the (a) original image, (b) CNN prediction, (c) Hough transform (blue line), and (d) ground truth (orange line).



(c) $\tau = 0.6 \cdot \max_{x,y} P(x, y)$

Fig. 9. Example by varying the threshold on branch detection. The orange and blue lines represent ground truth and prediction respectively.

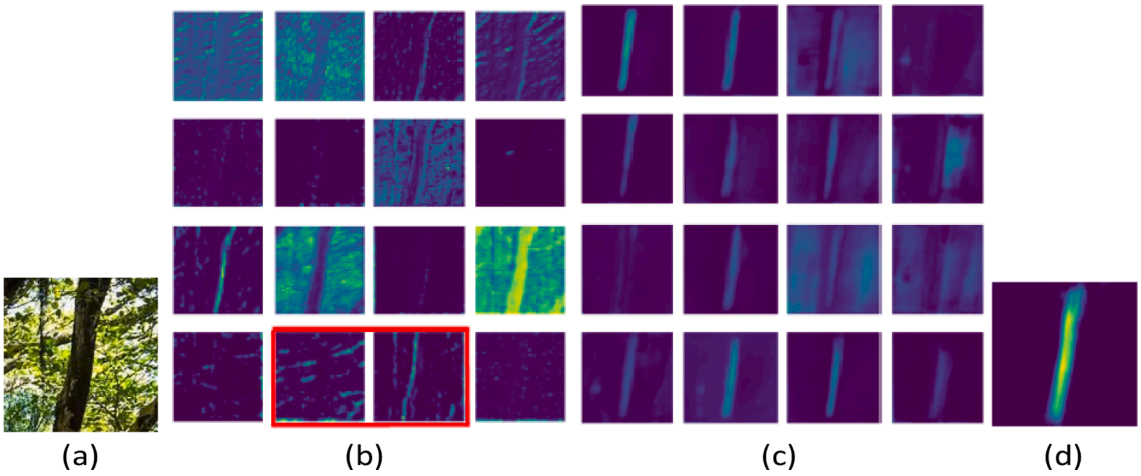


Fig. 10. Feature map of the (b) fourth layer, (c) tenth layer and (d) the last layer after processing the (a) input image containing a vertical branch.

probability of the points on the line, it is not necessarily at the center of the tree branch. This is important for tree branches occluding with other ones or even leaves. For example, the second row in Fig. 8 shows a tree branch in occlusion with another smaller one that could interfere with the grip. Even with these challenges, the proposed point is in a region without occlusion.

An important step in the proposed method is the variable threshold τ. Although the threshold adapts to each image, it does not guarantee that only points within the tree branch will be detected. However, outliers (if present) are the minority and the application of the Hough transform tends to ignore them and detect only the main line. Fig. 9 presents an example varying the threshold from 0.1, 0.35 (default) to 0.6. As we can see, even with the presence of outliers (Figs. 9a and 9b), the Hough transform was able to robustly detect the line.

To visualize the learning of the proposed method, Figs. 10 and 11 illustrate the feature map of the convolution layers after processing the input image with a vertical and horizontal tree branch. Understanding the final prediction of a CNN requires interpreting the activation of filters in the inner layers. In general, the early layer filters extract general features (e.g., Fig. 10b and 11b). Activation maps highlighted in red, for example, detect regions with vertical and horizontal edges. Other maps, like the one in the upper left, are activated on diagonal edges. On the

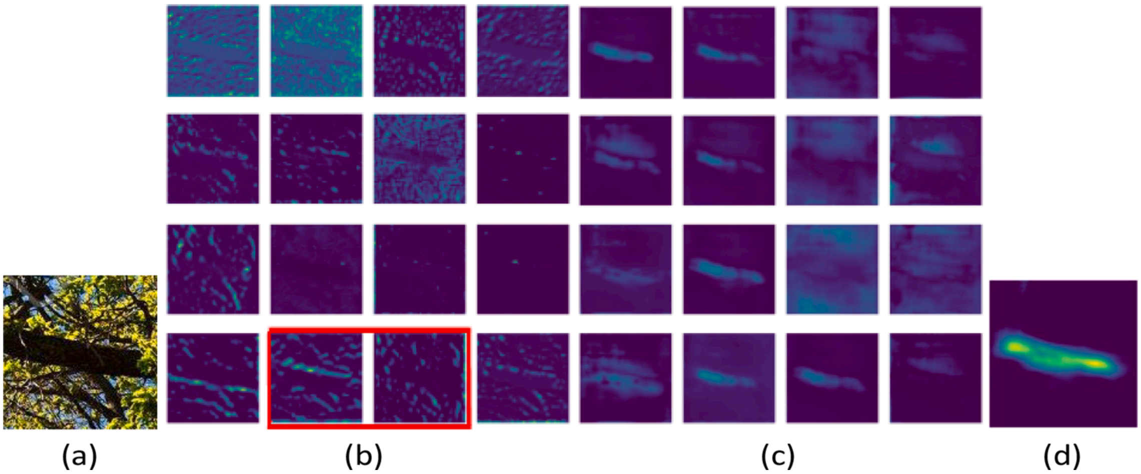


Fig. 11. Feature map of the (b) fourth layer, (c) tenth layer and (d) the last layer after processing the (a) input image containing a horizontal branch.

Table 3
Comparison (F1-score) between the proposed method, DHT, and F-Clip in Gaussian and Shot noise. Best results are in boldface.

Corruption	Severity	F-Clip	DHT	Ours Lightweight	Ours UNet
non-corruption	-	0.9371	0.9661	**0.9678**	**0.9684**
Gaussian	low	0.7700	**0.7943**	0.7138	**0.8748**
	high	**0.6794**	0.4653	0.6627	**0.7849**
Shot	low	0.7097	0.7564	**0.7931**	**0.9085**
	high	0.6963	0.5433	**0.7227**	**0.7994**

other hand, the deeper layers specialize in identifying the main branch and the regions around it (e.g., Figs. 10c and 11c). Due to the large number of examples during training, our method was able to generalize and recognize the main branch (see Figs. 10d and 11d).

3.3. Comparison with Literature Method

In this section, we compare the proposed method with the Deep Hough Transform (DHT) (Zhao et al., 2021) and Fully Convolutional Line Parsing (F-Clip) (Dai et al., 2021), two recent and state-of-the-art methods for line detection. For the proposed method, two versions were used: the lighter version with the proposed CNN and a heavier version with the UNet. DHT combines the traditional Hough transform with deep features into an end-to-end framework for detecting lines in images. F-Clip proposes a CNN to directly predict the center position, length and angle of the line in the image (We used version HG1-D2). Although DHT and F-Clip can be applied in edge detection, they are able to learn rich contextual semantics of lines. After training in our dataset, it proved to be robust in detecting tree branches, as our images also show natural scenes.

Table 3 shows the comparative results between the methods. The proposed method has a small improvement in relation to DHT, being 0.9678 (lightweight) and 0.9684 (UNet) against 0.9661 in the test images, as shown in the first line of the table. On the other hand, both versions of the proposed method were superior to the F-Clip. This shows that the proposed method is competitive with the state-of-the-art, even with a simple CNN with few layers.

Table 3 also presents the results for test images corrupted with two types of noise (Gaussian and Shot) and two severities (low and high). With Gaussian noise, the low and high severities correspond respectively to the Gaussian distribution with standard deviation equal to 0.08 and 0.12 (Hendrycks and Dietterich, 2019). On the other hand, Poisson distribution with 60 and 25 was used in the Shot noise for low and high severities, respectively (Hendrycks and Dietterich, 2019).

As expected, the noise affects tree branch detection in all methods. For the low severity in Gaussian noise, the proposed method with UNet obtained the best result followed by the DHT, an improvement from 0.7943 to 0.8748. The lightweight version of the proposed method also obtained relevant results due to its low computational cost compared to other methods. For high severity, the UNet version of the proposed method provided the best results. However,the proposed method in its lightweight version provided an improvement of approximately 0.2 in relation to DHT and competitive results compared to F-Clip.

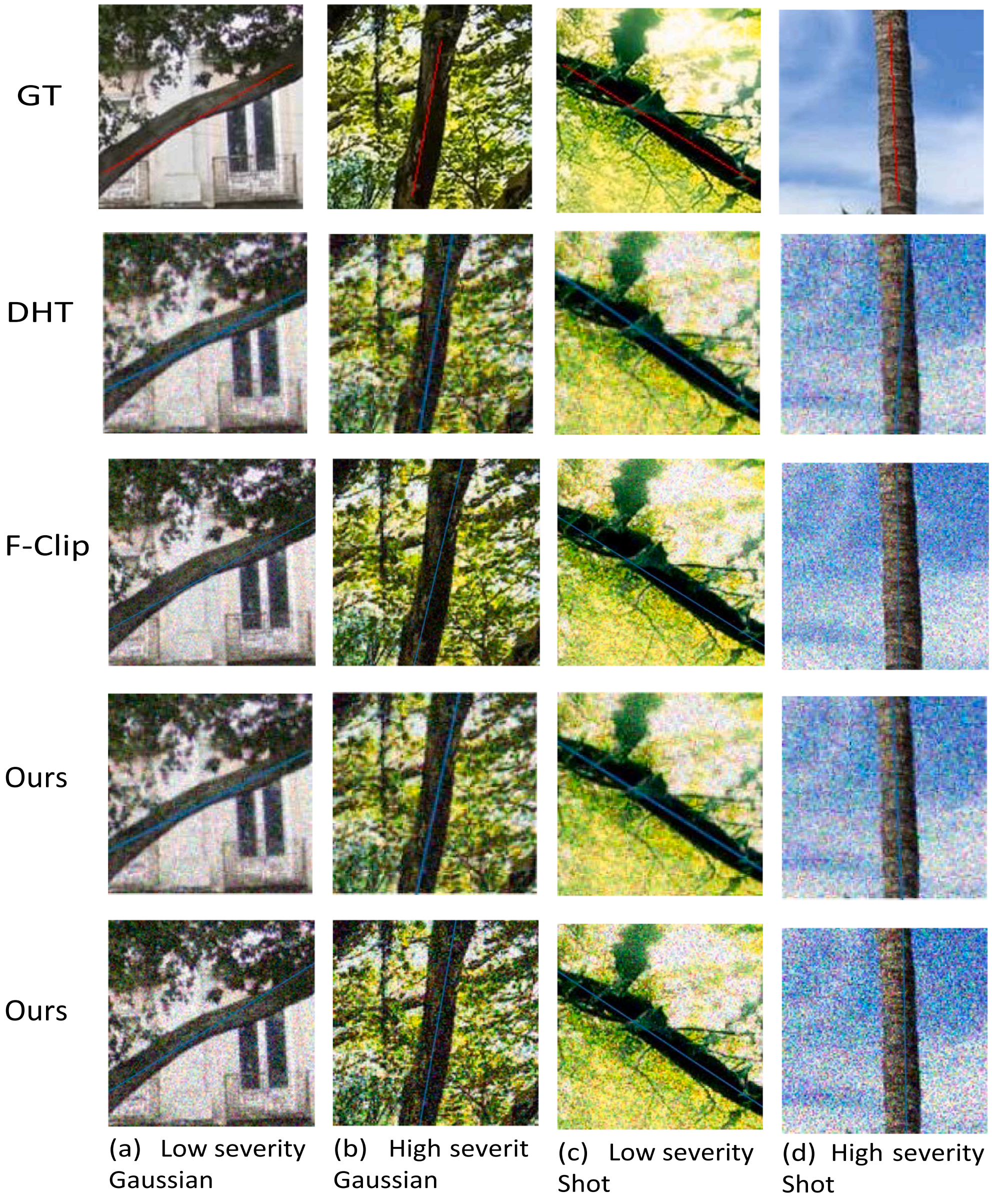
For Shot noise, the proposed method in its two versions were superior to DHT and F-Clip in both severities, improving from 0.7564 and 0.7097 to 0.9085 in low severity and from 0.5433 and 0.6963 to 0.7994 in high severity. Fig. 12 illustrates examples of tree branch detection with both noise and severities. These results show that the model presents strong robustness.

Finally, we evaluated the computational cost of methods for image inference. For that, we ran each of the methods on 50 images with 128 × 128 pixels and reported the average inference time for an image and its respective frames per second (FPS) in Table 4. For these experiments, we used colab which has a Tesla K80 GPU. As we can see, F-Clip is the fastest method with 60.86 frames per second, followed by DHT and our method.

3.4. Discussion

Tree branches can be a significant problem when reaching the power lines, causing electricity shortages. The cutting, in general, is done manually by specialists, introducing risks for them. They face limited vision due to overgrown branches and twigs, making their work even harder. Here, we propose a deep learning approach to support automatic branch pruning, focusing on the vision sensor. We assume a camera in a robotic arm positioned by an operator pointing near the branch of interest.

As we intend to perform the processing in loco, real-time solutions are required with high accuracy. The practical use of the methods must balance accuracy and efficiency. For this evaluation, Fig. 13 presents a plot of F1 versus time (ms) for all methods. F1 is given by the average of the tests with and without noise from Table 3. The methods closer to the lower right corner have higher F1 and lower inference time. We can see that the proposed method with UNet outperforms the others in terms of accuracy by a good margin; however, it is more computational expensive with an inference time of 52 ms (19 frames per second). Finally, we can verify from Fig. 13 that our lightweight model provides a good balance between accuracy and time.



(a) Low severity Gaussian
(b) High severit Gaussian
(c) Low severity Shot
(d) High severity Shot

Fig. 12. Comparison between DHT (second row), F-Clip (third row), our lightweight version (fourth row), and our UNet version (fifth row) for images with Gaussian and Shot noises at low and high severities. The ground truth is presented in the first row of images.

Table 4
Computational cost for image inference by the methods.

	F-Clip	DHT	Ours Lightweight	Ours UNet
Time (ms)	**16.43**	33.80	43.70	52.57
FPS	**60.86**	29.59	22.89	19.02

For future works, multi-scale fusion and attention mechanisms can be applied to improve the accuracy; however, it is important to be aware of the processing time, as it is very important in this application.

4. Conclusion

We proposed a deep learning-based method capable of obtaining the position and angulation of branches of a wide variety of species of trees in conjunction with a candidate grip point. We built a novel dataset of 1868 images with at least one main tree branch. We provide a quantitative analysis under different corruption scenarios and a qualitative analysis of the results. We also compared our results with the Deep Hough Transform (DHT), the state-of-the-art method for line detection.

Our results show that our model presents comparable performance to novel methods for line detection (DTH). Further, our model outperformed DTH for high severity corruption scenarios, showing the robustness of our model. We highlighted that our model is capable of proposing a candidate grip point. Furthermore, energy companies can benefit from this model, helping to make a safe and accurate predictive maintenance of the power grid. Our work can be used to build powerful tools helping companies to infinity sensitive areas of the power grid quickly and safely. We intend to adapt the method to detect multiple tree branches simultaneously aiming to advance in the procedure automation. We also intend to embed the proposed method on a device to run in real time, and manually label a suitable point for grasping and perform the prediction with CNN.

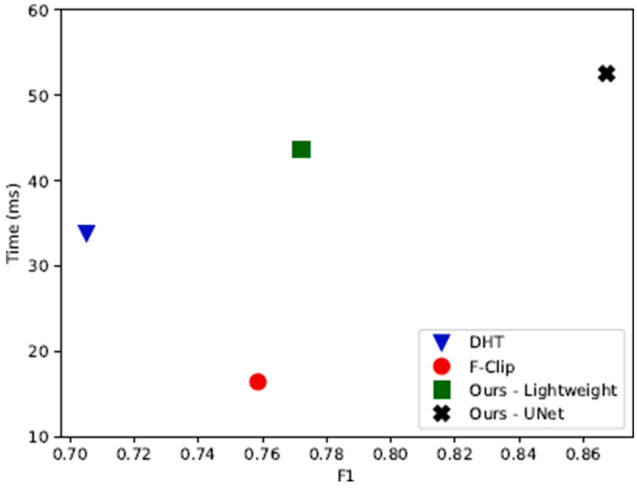


Fig. 13. Mean F1 versus time (ms) of the methods.

Funding

This research was funded by CNPq (p: 433783/2018–4, 310517/2020–6, 314902/2018–0, 304052/2019–1 and 303559/2019–5), FUNDECT (p: 59/300. 066/2015, 071/2015) and CAPES PrInt (p: 88881.311850/2018–01). The authors acknowledge the support of the UFMS (Federal University of Mato Grosso do Sul) and CAPES (Finance Code 001). This research was also partially supported by the Emerging Interdisciplinary Project of Central University of Finance and Economics.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to acknowledge Nvidia Corporation for the donation of the Titan X graphics card.

References

Alam, M.M., Zhu, Z., Tokgoz, B.E., Zhang, J., Hwang, S., 2020. Automatic assessment and prediction of the resilience of utility poles using unmanned aerial vehicles and computer vision techniques. Int. J. Disaster Risk Sci. 11, 119–132.

Amatya, S., Karkee, M., Gongal, A., Zhang, Q., Whiting, M.D., 2016. Detection of cherry tree branches with full foliage in planar architecture for automated sweet-cherry harvesting. Biosystems Engineering 146, 3–15. URL: https://www.sciencedirect.com/science/article/pii/S1537511015001683, doi: 10.1016/j.biosystemseng.2015.10.003. special Issue: Advances in Robotic Agriculture for Crops.

Brock, A., De, S., Smith, S.L., Simonyan, K., 2021. High-performance large-scale image recognition without normalization. CoRR abs/2102.06171. URL: https://arxiv.org/abs/2102.06171.

Chen, M., Tang, Y., Zou, X., Huang, Z., Zhou, H., Chen, S., 2021. 3d global mapping of large-scale unstructured orchard integrating eye-in-hand stereo vision and slam. Computers and Electronics in Agriculture 187, 106237. URL: https://www.sciencedirect.com/science/article/pii/S0168169921002544, doi: 10.1016/j.compag.2021.106237.

da Silva, L.A., a Bressan, P.O., Gonçalves, D.N., Freitas, D.M., Machado, B.B., Gonçalves, W.N., 2019. Estimating soybean leaf defoliation using convolutional neural networks and synthetic images. Computers and Electronics in Agriculture 156, 360–368. URL: https://www.sciencedirect.com/science/article/pii/S0168169918307907, doi: 10.1016/j.compag.2018.11.040.

Dai, X., Yuan, X., Gong, H., Ma, Y., 2021. Fully convolutional line parsing. arXiv: 2104.11207.

David Reiser, J.S., Griepentrog, H.W., 2021. Autonomer baumschnitt in streuobstwiesen. 41. GIL-Jahrestagung, Informations-und Kommunikationstechnologie in kritischen Zeiten.

Gomes, M., Silva, J., Gonçalves, D., Zamboni, P., Perez, J., Batista, E., Ramos, A., Osco, L., Matsubara, E., Li, J., et al., 2020. Mapping utility poles in aerial orthoimages using atss deep learning method. Sensors 20, 6070. https://doi.org/10.3390/s20216070.

Guan, J., Su, Y., Su, L., Sivaparthipan, C., Muthu, B., 2021. Bio-inspired algorithms for industrial robot control using deep learning methods. Sustainable Energy Technologies and Assessments 47, 101473. URL: https://www.sciencedirect.com/science/article/pii/S2213138821004835, doi: 10.1016/j.seta.2021.101473.

Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., Bennamoun, M., 2020. Deep learning for 3d point clouds: A survey. IEEE Trans. Pattern Anal. Mach. Intell. 1–1 https://doi.org/10.1109/TPAMI.2020.3005434.

He, L., Schupp, J., 2018. Sensing and automation in pruning of apple trees: A review. Agronomy 8. URL: https://www.mdpi.com/2073-4395/8/10/211, doi:10.3390/agronomy8100211.

Hendrycks, D., Dietterich, T., 2019. Benchmarking neural network robustness to common corruptions and perturbations. International Conference on Learning Representations (ICLR) 1–16.

Straub, Jonas, David Reiser, H.W.G., 2021. Approach for modeling single branches of meadow orchard trees with 3d point clouds. Precis. Agric. 21, 735–741.

Karkee, M., Adhikari, B., 2015. A method for three-dimensional reconstruction of apple trees for automated pruning. Trans. ASABE 58, 565–574.

Karkee, M., Adhikari, B., Amatya, S., Zhang, Q., 2014. Identification of pruning branches in tall spindle apple trees for automated pruning. Comput. Electron. Agric. 103, 127–135.

Lu, Z., Zhao, M., Luo, J., Wang, G., Wang, D., 2021. Design of a winter-jujube grading robot based on machine vision. Computers and Electronics in Agriculture 186, 106170. URL: https://www.sciencedirect.com/science/article/pii/S0168169921001873, doi: 10.1016/j.compag.2021.106170.

Luongo, F., Hakim, R., Nguyen, J.H., Anandkumar, A., Hung, A.J., 2021. Deep learning-based computer vision to recognize and classify suturing gestures in robot-assisted surgery. Surgery 169, 1240–1244. URL: https://www.sciencedirect.com/science/article/pii/S0039606020305481, doi: 10.1016/j.surg.2020.08.016.

Ma, L., Liu, Y., Zhang, X., Ye, Y., Yin, G., Johnson, B.A., 2019. Deep learning in remote sensing applications: A meta-analysis and review. ISPRS Journal of Photogrammetry and Remote Sensing 152, 166–177. URL: https://www.sciencedirect.com/science/article/pii/S0924271619301108, doi: 10.1016/j.isprsjprs.2019.04.015.

Majeed, Y., Zhang, J., Zhang, X., Fu, L., Karkee, M., Zhang, Q., Whiting, M.D., 2018. Apple tree trunk and branch segmentation for automatic trellis training using convolutional neural network based semantic segmentation. IFAC-PapersOnLine 51, 75–80.

Majeed, Y., Zhang, J., Zhang, X., Fu, L., Karkee, M., Zhang, Q., Whiting, M.D., 2020. Deep learning based segmentation for automated training of apple trees on trellis wires. Computers and Electronics in Agriculture 170, 105277. URL: https://www.sciencedirect.com/science/article/pii/S0168169919316266, doi: 10.1016/j.compag.2020.105277.

Molina, J., Hirai, S., 2017. A grasping-climbing mechanism for pruning tree-branches using a multirotor helicopter. The Proceedings of JSME annual Conference on Robotics and Mechatronics (Robomec) 2017, 1P1–F01. doi:10.1299/jsmermd.2017.1P1-F01.

Odo., A., McKenna., S., Flynn., D., Vorstius., J., 2020. Towards the automatic visual monitoring of electricity pylons from aerial images, in: Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5: VISAPP, INSTICC. SciTePress. pp. 566–573. doi:10.5220/0009345005660573.

Osco, L.P., dos Santos de Arruda, M., Gonçalves, D.N., Dias, A., Batistoti, J., de Souza, M., Gomes, F.D.G., Ramos, A.P.M., de Castro Jorge, L.A., Liesenberg, V., Li, J., Ma, L., Marcato, J., Gonçalves, W.N., 2021a. A cnn approach to simultaneously count plants and detect plantation-rows from uav imagery. ISPRS Journal of Photogrammetry and Remote Sensing 174, 1–17. URL: https://www.sciencedirect.com/science/article/pii/S0924271621000307, doi: 10.1016/j.isprsjprs.2021.01.024.

Osco, L.P., Marcato Junior, J., Marques Ramos, A.P., de Castro Jorge, L.A., Fatholahi, S. N., de Andrade Silva, J., Matsubara, E.T., Pistori, H., Gonçalves, W.N., Li, J., 2021b. A review on deep learning in uav remote sensing. International Journal of Applied Earth Observation and Geoinformation 102, 102456. URL: https://www.sciencedirect.com/science/article/pii/S030324342100163X, doi: 10.1016/j.jag.2021.102456.

Parent, J.R., Meyer, T.H., Volin, J.C., Fahey, R.T., Witharana, C., 2019. An analysis of enhanced tree trimming effectiveness on reducing power outages. J. Environ. Manage. 241, 397–406. https://doi.org/10.1016/j.jenvman.2019.04.027.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. Springer International Publishing, Cham, pp. 234–241.

Shafiei, S.B., Iqbal, U., Hussein, A.A., Guru, K.A., 2021. Utilizing deep neural networks and electroencephalogram for objective evaluation of surgeonś distraction during robot-assisted surgery. Brain Research 1769, 147607. URL: https://www.sciencedirect.com/science/article/pii/S0006899321004649, doi: 10.1016/j.brainres.2021.147607.

Siebert, L.C., Toledo, L.F.R.B., Block, P.A.B., Bahlke, D.B., Roncolatto, R.A., Cerqueira, D. P., 2014. A survey of applied robotics for tree pruning near overhead power lines. In: Proceedings of the 2014 3rd International Conference on Applied Robotics for the Power Industry, pp. 1–5. https://doi.org/10.1109/CARPI.2014.7030070.

Tang, Y., Chen, M., Wang, C., Luo, L., Li, J., Lian, G., Zou, X., 2020. Recognition and localization methods for vision-based fruit picking robots: A review. Front. Plant Sci. 19 https://doi.org/10.3389/fpls.2020.00510.

Wang, W., Tian, W., Liao, W., Li, B., 2021. Pose accuracy compensation of mobile industry robot with binocular vision measurement and deep belief network. Optik 238, 166716. URL: https://www.sciencedirect.com/science/article/pii/S0030402621004320, doi: 10.1016/j.ijleo.2021.166716.

Yang, H., Chen, L., Ma, Z., Chen, M., Zhong, Y., Deng, F., Li, M., 2021. Computer vision-based high-quality tea automatic plucking robot using delta parallel manipulator. Comput. Electron. Agric. 181, 105946. https://doi.org/10.1016/j.compag.2020.105946.

Yuan, X., Shi, J., Gu, L., 2021. A review of deep learning methods for semantic segmentation of remote sensing imagery. Expert Systems with Applications 169, 114417. URL: https://www.sciencedirect.com/science/article/pii/S0957417420310836, doi: 10.1016/j.eswa.2020.114417.

Zahid, A., Mahmud, M.S., He, L., Heinemann, P., Choi, D., Schupp, J., 2021. Technological advancements towards developing a robotic pruner for apple trees: A review. Computers and Electronics in Agriculture 189, 106383. URL: https://www.sciencedirect.com/science/article/pii/S0168169921004002, doi: 10.1016/j.compag.2021.106383.

Zamboni, P., Junior, J.M., Silva, J.d.A., Miyoshi, G.T., Matsubara, E.T., Nogueira, K., Gonçalves, W.N., 2021. Benchmarking anchor-based and anchor-free state-of-the-art deep learning methods for individual tree detection in rgb high-resolution images. Remote Sensing 13. URL: https://www.mdpi.com/2072-4292/13/13/2482, doi: 10.3390/rs13132482.

Zhang, J., He, L., Karkee, M., Zhang, Q., Zhang, X., Gao, Z., 2018a. Branch detection for apple trees trained in fruiting wall architecture using depth features and Regions-Convolutional Neural Network (R-CNN). Computers and Electronics in Agriculture 155, 386–393. URL: https://www.sciencedirect.com/science/article/pii/S0168169918304162, doi: 10.1016/j.compag.2018.10.029.

Zhang, W., Witharana, C., Li, W., Zhang, C., Li, X., Parent, J., 2018b. Using deep learning to identify utility poles with crossarms and estimate their locations from google street view images. Sensors 18. URL: https://www.mdpi.com/1424-8220/18/8/2484, doi:10.3390/s18082484.

Zhang, X., Karkee, M., Zhang, Q., Whiting, M.D., 2021. Computer vision-based tree trunk and branch identification and shaking points detection in dense-foliage canopy for automated harvesting of apples. J. Field Robot. 38, 476–493.

Zhao, K., Han, Q., Zhang, C.B., Xu, J., Cheng, M.M., 2021. Deep hough transform for semantic line detection. IEEE Trans. Pattern Anal. Mach. Intell. 1–1 https://doi.org/10.1109/TPAMI.2021.3077129.