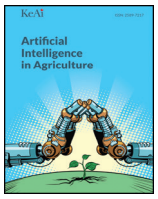
Artificial Intelligence in Agriculture 7 (2023) 13–26

Contents lists available at ScienceDirect

Artificial Intelligence in Agriculture

journal homepage: http://www.keaipublishing.com/en/journals/artificial-intelligence-in-agriculture/



Deep learning for the detection of semantic features in tree X-ray CT scans

Salim Khazem [a,b,*], Antoine Richard [e], Jeremy Fix [b,d], Cédric Pradalier [a,c]

[a] GeorgiaTech-CNRS IRL 2958, Metz, France
[b] CentraleSupélec, Metz, France
[c] GeorgiaTech Lorraine, Metz, France
[d] LORIA, CNRS UMR 7503, Metz, France
[e] University of Luxembourg, Campus Kirchberg, 6, rue Richard Coudenhove-Kalergi L-1359 Luxembourg

ARTICLE INFO

Article history:
Received 10 August 2022
Received in revised form 13 December 2022
Accepted 24 December 2022
Available online 4 January 2023

Keywords:
X-rays images
Deep learning
Convolutional neural networks
Image segmentation
Wood knots
Coordinates detection
Contour estimation

ABSTRACT

According to the industry, the value of wood logs is heavily influenced by their internal structure, particularly the distribution of knots within the trees. Nowadays, CT scanners combined with classical computer vision approach are the most common tool for obtaining reliable and accurate images of the interior structure of trees. Knowing where the tree semantic features, especially knots, contours and centers are within a tree could improve the efficiency of the overall tree industry by minimizing waste and enhancing the quality of wood-log by-products. However, this requires to automatically process the CT-scanner images so as to extract the different elements such as tree centerline, knot localization and log contour, in a robust and efficient manner. In this paper, we propose an effective methodology based on deep learning for performing these different tasks by processing CT-scanner images with deep convolutional neural networks. To meet this objective, three end-to-end trainable pipelines are proposed. The first pipeline is focused on centers detection using CNNs architecture with a regression head, the second and the third one address contour estimation and knot detection as a binary segmentation task based on an Encoder-Decoder architecture. The different architectures are tested on several tree species. With these experiments, we demonstrate that our approaches can be used to extract the different elements of trees in a precise manner while preserving good performances of robustness. The main objective was to demonstrate that methods based on deep learning might be used and have a relevant potential for segmentation and regression on CT-scans of tree trunks.

1. Introduction

Knots are one of the most significant factors in the wood processing chain. A knot is defined as the part of a branch embedded in the trunk that generally arises at the tip of the trunk (Longo et al., 2019). Biologically, all branches grow up from the pith, the center of the tree stem and the growth rings. The knots have a direct impact on the quality and value of logs, which makes knowing their exact localization and distribution within the logs relevant and crucial for foresters and sawyers. If the knot, defect location and size are known before sawing (Bhandarkar et al., 1999), this could generate a potential gain of 15–18% in value of products. Moreover, this knowledge would support forest scientist by providing an insight on the mechanisms of tree growth based on geometric measurements such as (knot length, ring diameter, etc.), without having to fell the tree.

Detection of centers, contours, and knots is a very relevant task for the wood industry, which can significantly improve production quality, and yield. Nowadays, sawmills address these needs by scanning tree logs with various sensors before planning how it will be cut. Among these sensors, 3D scanners build a 3D model of the tree exterior, mostly towards volume estimation, and X-ray Computed Tomography (CT) provides a 3D density model of the tree interior, represented as a stack of image describing slices orthogonal to the length of the tree. From such image stacks, tree semantic features (centerline, contour, knots) are localized using traditional computer vision methods (Krähenbühl et al., 2012, 2013a, 2014).

These traditional approaches have shown promising results on some species. However, they may lack robustness in challenging cases such as for detecting knots when the sapwood is wet (in these situations, the density of the wood is similar to the density of the knot). With the

* Corresponding author at: GeorgiaTech-CNRS IRL 2958, Metz, France.
E-mail address: skhazem@georgiatech-metz.fr (S. Khazem).

https://doi.org/10.1016/j.aiia.2022.12.001

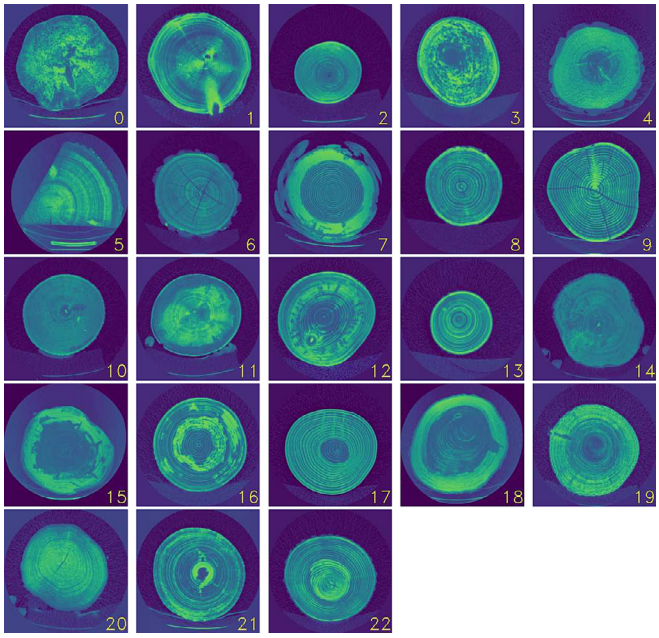


Fig. 1. Illustration of the different tree species contained in the dataset with respective indexes (see Table 1).

recent advent of deep learning and its success on a variety of machine learning problems, the limitations of traditional approaches may be overcome, although at the cost of requiring a larger amount of annotated data.

In this paper, we demonstrate how data driven machine learning and specifically deep learning, can be used to address these robustness issues on three specific applications:

- For tree centerline detection, we train a regression model to predict the intersection of the tree centerline with each X-ray CT images (for a given slice, we refer to this intersection as the tree center). In comparison to previous studies, this is tested on a large set of species and has shown its robustness.
- Using an Encoder-Decoder architecture, we demonstrate how tree contour extraction can be reliably cast as an image segmentation problem. Here as well, we tested on a large set of species than previous studies, and we demonstrate its robustness.
- Finally, using the same segmentation network, we showcase the detection of knots in the CT images and validate it on a subset of species for which knots have been labeled by hand. This allows us to highlight the robustness challenges and the performance of our approach.

All the results presented in this paper have been evaluated on a tree database containing CT-scans of 682 trees from 23 species (see Fig. 1). Fig. 2 illustrates the number of trees by species and the number of images by species according to the indexes reported on Table 1.

2. Related work

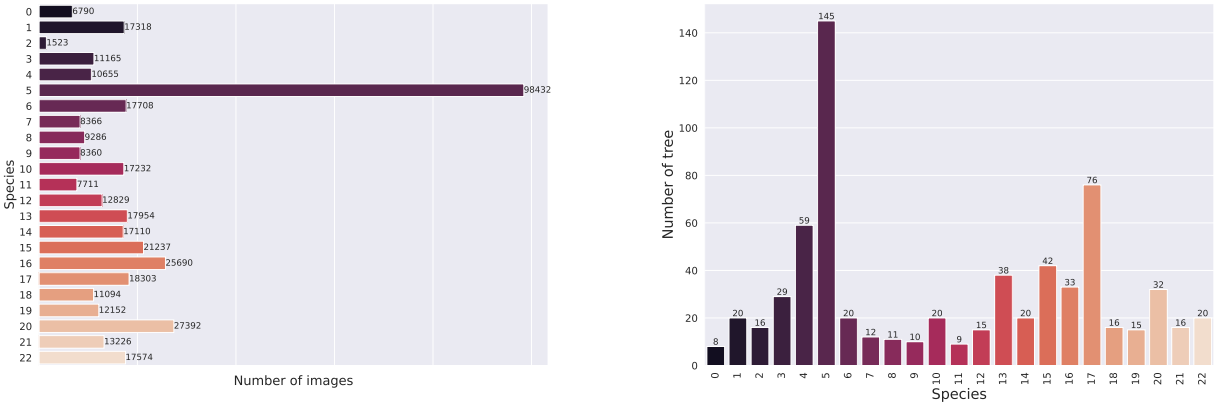
This paper can be compared with two categories of related work. On the one hand, a significant body of work in the deep learning community addresses questions of regression from images, image segmentation or detection of objects. However, these tools have rarely been applied to support the wood processing industry. On the other hand, we identified a number of works applied specifically to wood imagery, either using traditional imaging techniques on planks or trunks or X-ray tomography.

2.1. The deep learning toolbox for image processing

Deep learning approaches have shown considerable promise in various tasks that involve handling massive volumes of digital data. In the field of computer vision, deep learning methods are also rapidly being applied in a wide range of area and have demonstrated remarkable improvement in several tasks such as image reconstruction, object detection, image classification and image segmentation. During the past decade, many approaches and architectures have been developed in this field such as ResNet-50 (He et al., 2016), MobileNets (Howard et al., 2017), EfficientNet (Tan and Le, 2019). These networks have been introduced for classification tasks, but can also be used for regression and as backbone for segmentation tasks. For the different segmentation tasks such as Semantic segmentation which is based on pixel level classification into a predefined set of classes or Instance segmentation which consists of detect, segment and classify each individual object. Some architectures such as UNet, SegNet, LinkNet, MaskRCNN and Upsnet (Ronneberger et al., 2015; Badrinarayanan et al., 2016; Chaurasia and Culurciello, 2017; He et al., 2017; Xiong et al., 2019) have shown great performances.

2.1.1. Network architectures

A deep learning architecture is defined as a multi-layers stack of basic modules which have the ability to learn to transform their input to improve the representation selectivity and invariance (LeCun et al., 2015). In the computer vision field, many deep learning architectures have evolved over the last few years. In particular, architectures based



(a) Number of images for each specie (b) Number of tree for each specie.

Fig. 2. Overview of the distribution of the number of tree per species and the number of Xray images per species.

Table 1
Species indexes.

Scientific name	English name	Index	Scientific name	English name	Index
Carpinus	Hornbeam	0	Betula	Birch	12
Prunus avium	Gean	1	Fraxinus	Ash	13
Acer	Maple	2	*Acer campestre*	Hedge Maple	14
Sorbus torminalis	Whitebeam	3	*Picea abies*	Spruce	15
Pinus sylvestris	Scots fir	4	Abies	Pine	16
Quercus	Oak	5	*Fagus sylvatica*	Beech	17
Acacia	Acacia	6	*Larix decidua*	Larch	18
Pseudotsuga menziesii	Douglas fir	7	Tilia	Lime	19
Alnus	Alder	8	*Quercus petraea*	Sessile oak	20
Quercus rubra	Red oak	9	*Populus tremula*	Aspen	21
Acer platanoides	Norway maple	10	Ulmus	Elm	22
Acer pseudoplatanus	Sycamore Maple	11	/	/	/

on Convolutional Neural Networks (CNNs) such as VGG, Residual network (ResNet), MobileNets and EfficientNet (Simonyan and Zisserman, 2015; He et al., 2016; Howard et al., 2017; Tan and Le, 2019) are developed to solve problems in different domains or use-cases and have significantly driven the performance of vision tasks based on their rich representation power. This led to the advent of more complex neural network such as (Liu et al., 2022) and those based on attention such as Vision Transformer (ViT), Data-Efficient Image Transformers (DeiT) and Swin Transformer (Dosovitskiy et al., 2021; Touvron et al., 2021; Liu et al., 2021).

2.1.2. Network losses

The training of a deep neural network is tightly specified by the loss function used to quantify the quality of the network's prediction with respect to and what it should have predicted, named label, ground-truth or target depending on the applications. In the case of a network performing a regression task, the typical loss is the mean-squared error defined below, where M is the number of dimension of the network output, y_i the target value and p_i the output of the network:

$$L_{MSE} = \frac{1}{M}\sum_{i=1}^{M}(y_i - p_i)^2 \tag{1}$$

For binary classification problems, the binary cross-entropy between the predicted probability of being positive and the ground truth, as defined in Eq. (2), is typically selected as the loss function to be minimized during the network training.

$$L_{BCE} = -\frac{1}{M}\sum_{i=1}^{M} y_i \log(p_i) + (1 - y_i)\log(1 - p_i) \tag{2}$$

Additionally, when the classes are unbalanced, the BCE is usually not sufficient to learn a good predictor and training may favor the majority classes. In such case, other losses such as the Dice loss might be preferred. The Dice loss function (Sudre et al., 2017), defined in Eq. (5), which tends to focus on pixels that are mispredicted by the model can also be included to supervise the model's training.

$$L_{Dice} = 1 - \frac{2\sum_{i=1}^{M} p_i y_i + \varepsilon}{\sum_{i=1}^{M} p_i + \sum_{i=1}^{M} y_i + \varepsilon} \tag{3}$$

In the context of an image segmentation task, for example for the prediction of the presence of knots, in Eqs. (2) and (3), M denotes the total number of pixels in the image and $y_i \in \{0,1\}$ represents the label of the i-th pixel, where 0 indicates that the pixel belongs to the background and 1 means that the pixel belongs to a knot. The probability of the segmentation model to predict a pixel belongs to a knot is denoted $p_i \in [0,1]$ and ε is a smoothness coefficient. In the different experiments, we empirically set $\varepsilon = 1$.

2.1.3. Data augmentation

Data augmentation is a commonly used technique for artificially increasing the size of the sample databases by applying different transformations to the available pair (images and labels). It increases the size of the input and creates diversity in the data. If correctly calibrated, data augmentation can improve generalization and performance of the training model (Shorten and Khoshgoftaar, 2019; Perez and Wang, 2017). In Computer vision, image augmentation has become a common implicit regularization approach to prevent overfitting (poor generalization) and generally enhances performances.

2.2. Detecting semantic features in trees

In the one hand, traditional optical imaging technique are still used in industry to perform processing and tasks such as object detection, classification and segmentation. In the wood industry, when it comes to detect defects on the surface of either the tree or planks, we usually use this kind of data due to their ease of acquisition and their practical use. In this context, we have identified some previous work using this type of data to detect defects and surface knots using deep learning based approaches (Gao et al., 2021a; Lopes et al., 2020; Norlander et al., 2015; Gao et al., 2021b). In the other hand, X-ray tomography represents one of the standard medical imaging modality which is considered as the most efficient way to get accurate and informative images of the inner structure non-invasively. In recent years, several industries, in particular the wood processing industry, have chosen to take advantage of this technology to obtain better understanding of their products in order to increase yield. Several approaches that aim to detect the inner properties and features of the tree have been proposed based on this modality (Kerautret and Lachaud, 2009; Krähenbühl et al., 2012, 2013a) and will be discussed below.

2.2.1. Expert methods

Tree semantic features (centers, knots, contours) within log cross-sections are an active area of research where high variability in knot appearance and labeled data availability are major challenges. Methods, such as (Johansson et al., 2013; Krähenbühl et al., 2013b) are developed to address this challenge. Johansson et al. (2013) is based on modeling the knot by non-concentric ellipses inside the log with different radius to localize the knots. They propose an ellipse detection algorithm that involves the following steps. First, they try to detect the heartwood Concentric Surface (CS), which is considered similar to a cylindrical shell, by fitting ellipses to the annual rings of the log. Then, all the objects (regions with high density value) on the concentric heartwood surface are fitted to ellipses and finally all overlapping ellipses across the heartwood concentric surface are tracked until reaching a sapwood CS which is considered as the end of the knot. (Krähenbühl et al., 2013a) proposed an approach to tackle the problem of segmenting wet area: they first detect the knot areas using (Krähenbühl et al., 2012) approach, and then exploit the geometrical properties such as dominant points and curvature computed from discrete contours (Kerautret and Lachaud, 2009) to estimate the position and orientation of the knots in the sapwood. Finally, this information is used to report a segmentation mask. However, these methods heavily rely on user expertise to adjust the parameters of the algorithms correctly. Let us take as example knot detection. Knots can have varied shapes that is hard to delineate. In addition, taking into the tree type, which its specific density, is not an easy task.

Indeed, the expert methods have been experimented only on a few tree species. In comparison, we are proposing end-to-end trained pipelines based on deep learning to address these different tree semantic applications, and we consider a much more diverse set of species.

2.2.2. Data driven methods

With the recent successes of neural networks on a wide range of machine learning problems, these techniques have also been experimented on wood logs CT scans. The work of (Norlander et al., 2015) showed that deep learning approaches, and specifically convolutional neural networks, outperformed a commercial detector based on feature descriptors and Support Vector Machine (SVM) (Cortes and Vapnik, 1995) when the task was to detect knots in oak tree planks. According to Norlander et al. (2015), the appropriate approach is to rely on neural networks architectures instead of relying on the traditional computer vision methods usually used in this field to locate the knots inside the CT-Scanned wood-logs. Several approaches are also proposed to detect knots on the surface of wood. Gao et al. (2021a) proposed an approach based on transfer learning using residual neural networks to detect defects (knots) on the surface of wood and perform a classification task to categorize them into seven defect types. The work of Lopes et al. (2020) consists of using and training The"You only look once" (Yolov3) (Redmon and Farhadi, 2018) architecture from scratch to perform knots detection on the surface of wood at high speed and have shown good results on knots that are on the surface. However, this approach has not been tested either on X-ray images or the inner knots which are very challenging.

Although many different architectures and models are available (see 2.1.1), their application in the wood industry remains limited and could be improved. Our work can be split into two main tasks: Regression and Segmentation. Pith estimation from CT scans is a regression task from images, and several end-to-end trainable neural networks mentioned in (2.1.1) with a regression head can be trained on this task. Knot detection can be either formulated as a segmentation task or object detection task, and there exists also a variety of end-to-end trainable neural networks in the computer vision community for solving this task. Contour estimation, on the other hand, can be addressed in several ways. One approach is to cast the contour estimation as tree segmentation where the task is to predict all the pixels belonging to the tree, hence a binary semantic segmentation task from which U-Net, Faster-RCNN, and any semantic segmentation neural network can be applied. That first approach, although classical, suffers from the difficulty of producing an output that is not necessarily a closed curve as expected for a contour predictor. A second approach relies on end-to-end trainable predictors outputting a closed curve. This is not straightforward to design a trainable neural network for outputting a closed curve, but recent works on differentiable active contours follow that track (Marcos et al., 2018). This is a recently proposed approach which has the benefit to output a closed curve, by constraint, but that we did not explore yet in this paper.

2.3. Synthesis on the related work

The various techniques described above highlight both the availability of a large and well-proven toolbox of deep-learning-based robust image processing techniques, and the existence of a number of image processing challenges whose solution would support the wood industry. This paper builds on these two observations to deploy and evaluate deep-learning solutions to the specific context of processing X-ray tomography of wood logs.

3. Materials and methods

Within this section we will cover different wood-logs semantic features extractions and propose techniques and models for the different tasks such as centerline detection, contour extraction, and knot segmentation.

3.1. Tree centerline detection

Within this part, we will present the proposed architecture for centers detection and the data augmentation strategy we selected.

Instead of proposing a new architecture, we have exploited the knowledge of a set of already existing convolutional neural network architectures (see 2.1.1) that have performed excellently on several image tasks. The choice of convolution networks is driven by the fact that our task consists on performing regression on images. For our work, we use ResNet-34 (He et al., 2016) which is a variant of the residual neural network with a total of 34 layers, the advantage of this architecture is that it offers a good combination of number of parameters and performance, and has proven its performance on image classification tasks. Another reason for exploiting the residual network architecture is the possibility of feeding images of different sizes from those with which they are trained thanks to global average pooling layer,[1] which computes the average value over the spatial dimensions of each features then feeded directly to the softmax activation. This is considered a crucial part to do transfer learning, which help us to achieve high performance on a network using very few epochs. The ResNet-34 network contains 21.2 million of trainable parameters.

Detecting the centers in the wood-log tomography slices is a pixel-coordinate regression, since the labels of the images are the center coordinates X and Y. Our work attempts to automate this task with an end-to-end trainable pipeline.

We trained a ResNet34 (He et al., 2016) on 256×256 pixel images with a regression head (the last linear layer pretrained on ImageNet is replaced with a linear layer with two units, initialized using the default pytorch strategy with both the weights and biases sampled from $\mathcal{U}\left(-\frac{1}{\sqrt{fan\_in}}, \frac{1}{\sqrt{fan\_in}}\right)$) and a mean-squared error loss (Eq. (1)). For regularization, we tested some data augmentation strategies (see 2.1.3) and early stopping to prevent overfitting. To ensure that the augmentation of these images is achievable with reasonable memory usage and speed, we used an open source library, Albumentations (Buslaev et al., 2020). The images are normalized between the values $[0, 1]$. We trained the network with the convolutional part pretrained on ImageNet, the first convolutional layer[2] being randomly initialized using the He normal strategy and the linear layer is randomly initialized using the default pytorch strategy with both the weights and biases sampled from $\mathcal{U}\left(-\frac{1}{\sqrt{fan\_in}}, \frac{1}{\sqrt{fan\_in}}\right)$. We used Adam (Kingma and Ba, 2015) as an optimizer with a learning rate $1e-3$ and the learning rate is reduced by a factor of 0.1 every time the validation metric stalls. We used a batch size of 256. We also used mixed precision training (Micikevicius et al., 2017) which sped up training without loss of performance. The loss function used in implementation is the Mean Squared Error (MSE) loss. We split the dataset into training and validation folds of, respectively, 90% and 10%. Fig. 3 illustrates our detailed regression pipeline for center detection.

For the data augmentation (seen in subsection 2.1.3), after testing several ones, we selected the following relevant transformations: shift, scale, color augmentation (saturation, brightness, contrast), random horizontal and vertical flipping, and rotations. We used the augmentation only with the training data. At the end, we did some experiments that consist of augmenting test data to evaluate the robustness of the model. The random seed of the data augmentation is fixed for reproducibility.

[1] It might appear as surprising that a network with global average pooling is able to predict a spatialized output but the empirical results actually show that this works and this architecture is significantly less parametrized than the same without the global average pooling layer.

[2] The networks pretrained on ImageNet involve color images with 3 inputs channels while our data involve only one channel, hence the first convolutional layer is replaced with convolutional kernels expected one input channel.

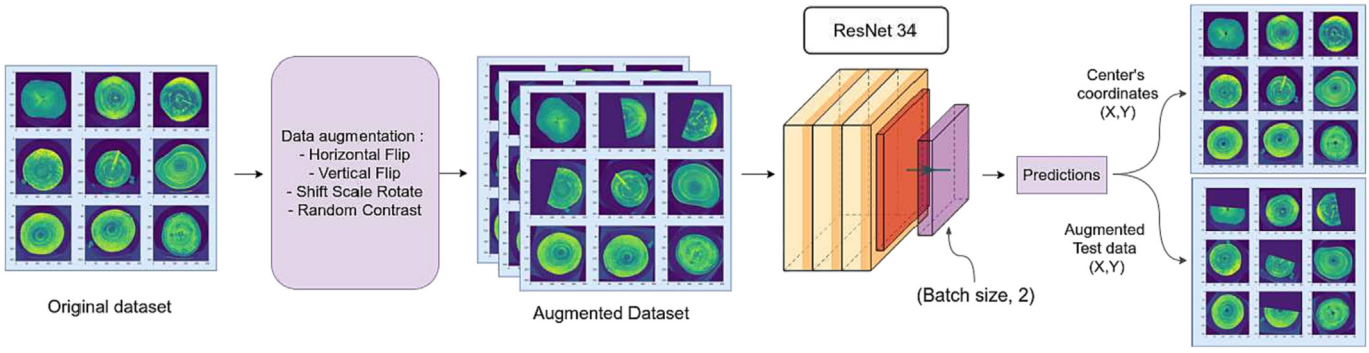


Fig. 3. The pipeline for regressing the tree center is based on a ResNet-34 convolutional backbone, pretrained on ImageNet, and a randomly initialized first convolutional layer and regression head. Training is performed on augmented data, applying random transformations to the training images and targets.

3.2. Trees contour detection

This part deals with the prediction of the contour of the tree. The contour is a thin line separating the wood log from the rest (background, wedges, etc…) in the CT-scan images. Rather than predicting that thin line, we transform the machine learning problem into the segmentation of the wood log from the rest, from which we can derive the contour.

We propose to use an Encoder-Decoder based model, and specifically U-Net (Ronneberger et al., 2015). In the encoder part, the model takes an image as input and applies a sequence of convolutions, max-pooling and ReLU activation and compresses the image into a latent space while extracting the most relevant features. The decoder part is a sequence of convolutional and transposed convolutional layers that attempt to decode the segmentation mask from the latent space and input image. Skip connections are used to take advantage of high-resolution layers of the encoder by sending information to the corresponding layers of the decoder, which allows the model to use fine-grained details learned from the encoder part to construct the image on the decoder part. These connections help the network to better capture small details that are present in high-resolution. The parameters of all the convolutional layers of the network are randomly initialized from $\mathcal{U}\left(-\frac{1}{\sqrt{fan\_in}}, \frac{1}{\sqrt{fan\_in}}\right)$. Before training, the first stage of our pipeline performs pre-processing on 512×512 pixels images by normalizing them in [0,1]. We use some on-the-fly non-destructive augmentation scheme to expand the dataset's size and improve the variability of samples, including random rotation and horizontal, vertical flip with a probability $p = 0.5$. The second stage consists of feeding the preprocessed data to the U-Net architecture and performing training. The loss function used for the U-Net is binary cross entropy[3] (Eq. (2)). We used RMSprop as an optimizer with a learning rate $1e-3$, a momentum 0.9 and a batch size of 8. The learning rate is decreased by a factor of 0.1 every time the validation error reaches a plateau. The data are split into a training fold of 90% and a validation fold of 10%. In the fold construction, it is possible that among consecutive slices, one is present in the training fold and the other in the validation fold, which may induce a bias in the estimation of the real risk from the validation fold. However, as will be shown in the results, the selected network which minimizes the validation loss is still able to generalize to unseen trees, even from unseen species. To prevent overfitting, in addition to the augmentation strategy, and weight decay of 10^{-8}, we used early stopping by selecting the best model as the one minimizing the BCE loss on the validation fold. The dataset used for the training is composed of 9 Abies (Fir) tree which represents 2504 annotated images. The pipeline for contour segmentation is illustrated in Fig. 4. More details of the architecture are illustrated in appendix Appendix C.

3.3. Tree knots detection

In this part, we present our approach to perform the knot segmentation task as binary mask prediction from single channel 512×512 CT-scan images.

The segmentation task requires separating the foreground (knots) and the background (other tissues). The network architecture is the same U-Net architecture used for the segmentation task in section 3.2. However, in addition to the binary cross entropy loss (Eq. (2)) and given that the wood slices present very few knots, we additionally use a Dice loss (Eq. (5)) to try to overcome class unbalancing. To handle the limited number of samples issue, we used a simple data augmentation strategy by applying horizontal and vertical flips on the fly. The dataset is split into training and validation folds of, respectively, 90% and 10 %. For the training parameters, we used the same as in the segmentation task in section 3.2. Fig. 5 depicts our detailed proposed pipeline to perform the knot segmentation task.

4. Experiments and comparisons

In this section, we present the various experiments and the results of the different tasks on stems of several tree species.

4.1. Datasets

4.1.1. CT images

The used images come from X-ray CT scanners in Digital Imaging and Communication in Medicine (**DICOM**) format which is considered the most popular standard in medicine, which makes medical image exchange easier and more independent of the imaging equipment manufacturer. In addition to the image, the DICOM format can also support other useful information to best describe the image such as width, height, in addition to some details related to the acquisition such as, technology used, serial number, date of acquisition (Mustra et al., 2008). The 512×512 pixels images are extracted from the DICOMs for the three tasks of center detection, contour and knots segmentation. Some sample images with their apparent knots are shown in Fig. 6. The Figs. 6 show X-ray images containing from 1 to 5 knots.

4.1.2. Annotations

The dataset has been manually annotated for the different tasks by ourselves. For the center detection task, the labels are provided in the form of X, Y floating coordinates of the wood-logs biological centers. For the segmentation tasks (eq. knots and contours) the labels are provided in the form of binary masks, each wood-log is associated with its related binary mask.

[3] In this task, on every slice, positive and negative samples are approximately balanced, hence a BCE loss makes sense.

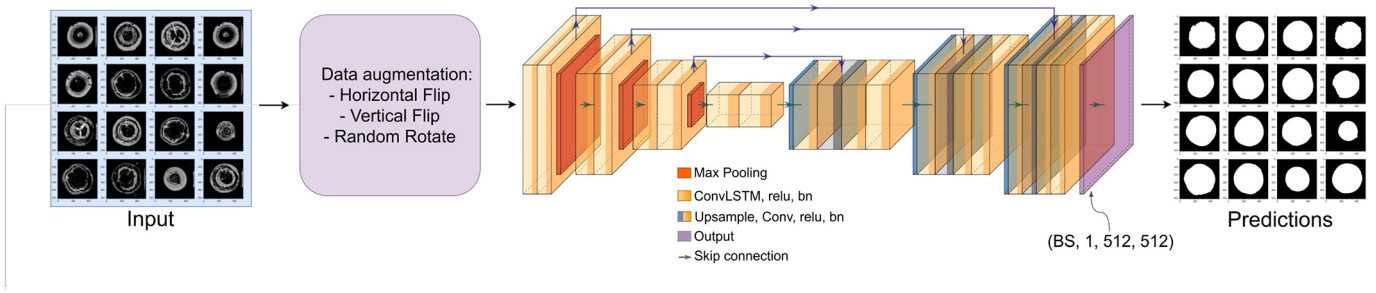


Fig. 4. The pipeline for the tree contour segmentation is based on an Encoder-Decoder U-Net architecture. Training is performed on augmented data, applying random non-destructive transformation to the training images and binary masks.

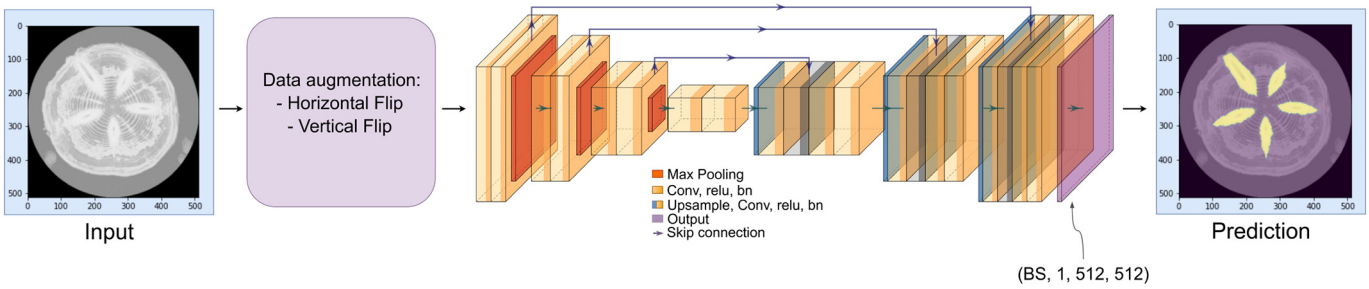


Fig. 5. Pipeline for the tree knots segmentation is based on an Encoder-Decoder U-Net architecture. Training is performed on augmented data, applying random non-destructive transformation to the training images and binary masks. The image on the right of the pipeline illustrates a prediction (yellow mask) overlayed with the input.

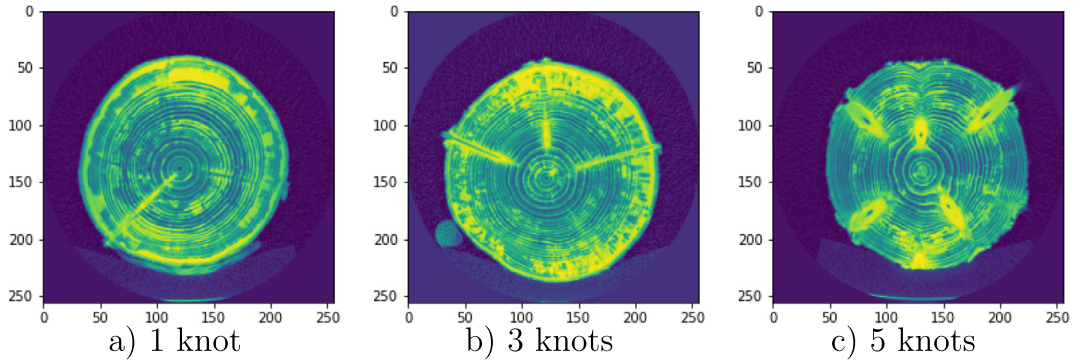


Fig. 6. Examples of used X-ray images and knot configuration with a color coding representing the wood density.(6-a) shows an example of a configuration which contains 1 knot.(6-b) illustrates an example with 3 knots which contain noise caused by the wet sapwood and (6-c) shows an example with 5 knots which also contain some noise cause by the wet sapwood.

4.2. Implementation details

We used Neptune[4] and Tensorboard[5] platforms to track the experiments and log the curves (mean absolute error, cross entropy loss and Dice loss). We used several workstations with an NVIDIA RTX 3090 GPU, an AMD Ryzen 5950 × 16 cores and 32 threads for training and inference.

The entire data preprocessing, the network models involved in the experiment, as well as the network training, were programmed in Python and Pytorch (Paszke et al., 2019) with Nvidia CUDA.[6] The center detection task was conducted for 40 epochs, The segmentation tasks (contour and knots) were conducted for 100 epochs per model.

[4] https://neptune.ai
[5] https://www.tensorflow.org/tensorboard
[6] Compute Unified Device Architecture

4.3. Metrics

For the several experiments, we used different quantitative metrics to evaluate the quality and performance of our approaches. For the regression task (centers detection) we use the Mean Absolute Error

Table 2
Model comparison with different parameters. The metrics are evaluated on the validation fold.

Augmentation	MAE (px)	MAE (mm)
/	6.6	6.58
Random Contrast	3.9	3.89
Vertical Flip	3.1	3.09
Horizontal Flip	2.3	2.29
Shift Scale Rotation	1.2	1.19
All	1.1	1.09

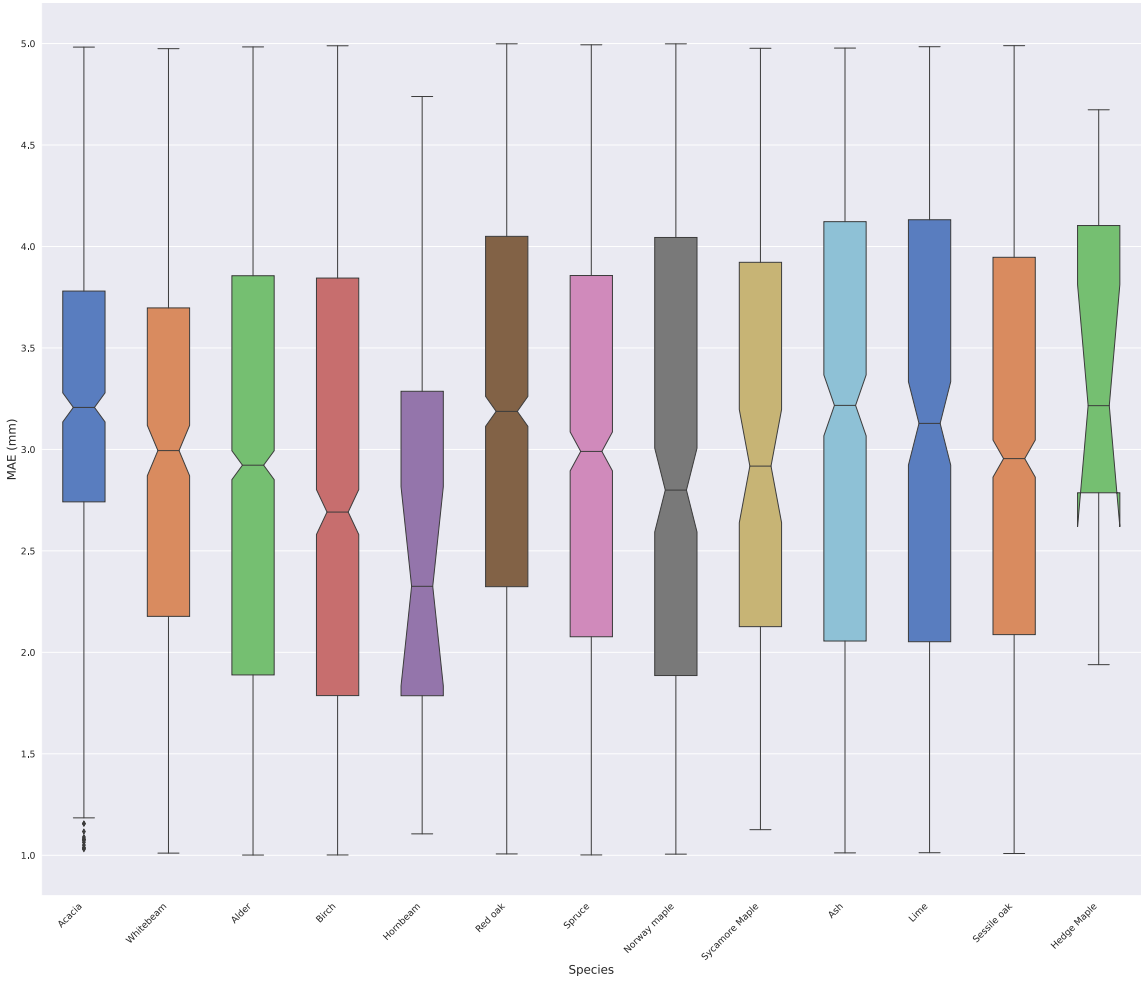


Fig. 7. Overview of the results of the model for different species for 8000 samples.

(MAE) as a metric to measures the quality of the predictor. For segmentation tasks (contour and knots), we evaluate the Dice coefficient (Dice), Accuracy and Mean Intersection Over Union (mean IoU). We denoted $\hat{y}$ as final prediction and y the true value to be predicted. For the segmentation tasks, we denote TP, TN, FP, FN respectively the number of true positives, true negatives, false positives and false negatives between y and $\hat{y}$. All the evaluation metrics adopted in our experiments could be formulated as follows:

4.3.1. Mean absolute error (MAE)

$$\forall y, \hat{y} \in \mathbb{R}, MAE(y, \hat{y}) = \mid y - \hat{y} \mid \tag{4}$$

4.3.2. Dice score / F1 score

$$\forall y, \hat{y} \in \{0,1\}^N, Dice(y, \hat{y}) = \frac{2TP(y, \hat{y})}{2TP(y, \hat{y}) + FP(y, \hat{y}) + FN(y, \hat{y})} \tag{5}$$

4.3.3. Mean intersection over Union (mean IoU)

$$\forall y, \hat{y} \in \{0,1\}^N, IoU(y, \hat{y}) = \frac{TP(y, \hat{y})}{TP(y, \hat{y}) + FP(y, \hat{y}) + FN(y, \hat{y})} \tag{6}$$

4.4. Results and discussions

4.4.1. Center detection

In this section, we present the results of the network performance on the tree centerline detection task, as well as the quantitative metrics we used. Table 2 presents the results of performances on center detection task with different parameters.

As shown in Table 2, our network achieves a good MAE for the different parameters we evaluated. In particular, we tested different types of augmentations individually to see the effect of each one on the performance of the model. We can observe that the data augmentation techniques improved considerably the performance of the networks. It can be also observed that Shift-Scale-Rotation and Horizontal Flip which are spatial-level transforms have more effect on the performance than pixel-level transform such as Random Contrast. Nevertheless, by combining all the proposed augmentations, we achieved a better performance with a **MAE** of **1.1 pixel** on the valid dataset, which correspond to an error of **2.2 mm**. The MAE metric is calculated on the original image (256×256).

To better highlight the robustness of the model to the different species, we tested our model with several species. The Fig. 7 presents the quantitative result for each specie. We noticed that except the little variability of the precision between species, the experience reveals that the model is globally robust. We also evaluated a simple predictor that always output the center of the image, independently of the species. We obtained an error of 7.88 pixel compared to 2.84 pixel for our model.

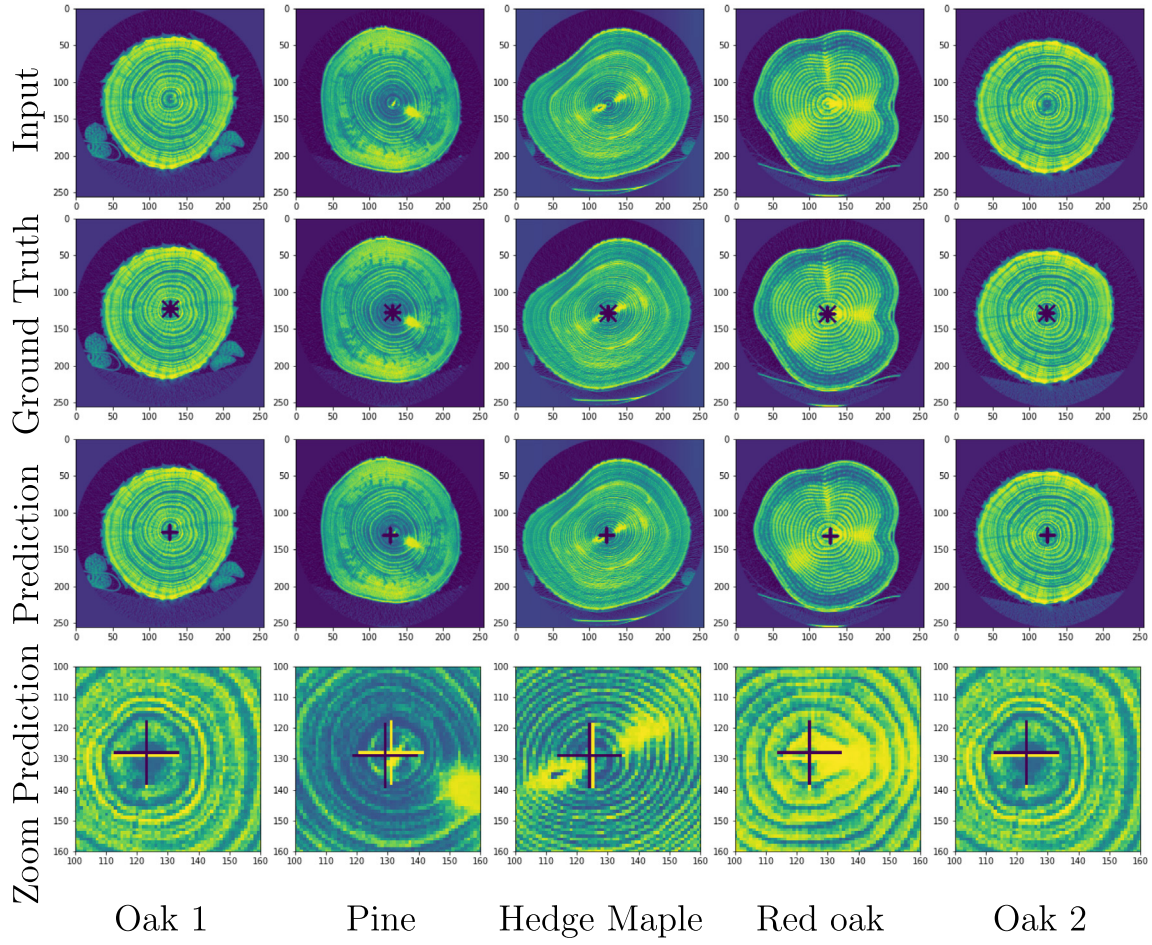


Fig. 8. Qualitative analysis of our model with different trees. The first row corresponds to the input images, the second row is the associated ground truth, the third one is the predictions and the last one illustrates a zoom in on the predictions where the blue cross represents the prediction and the yellow is the ground truth.

Additionally, we tested the impact of using AMP (Automated Mixed Precision) on the performance of the model, and we noticed that the difference in performance is not significant. For simple models, we obtained 6.1 pixel (6.09*mm*) without AMP and 6.6 pixel (6.58*mm*) with AMP. However, when using AMP, the gain of training time and the use of memory is significant. This is consistent with the work of (Micikevicius et al., 2018), which showed that the use of AMP leads to speedups of a factor 2-6× compared to the traditional method (FP32) with less memory consumption. As a result, all the evaluations presented in this paper use AMP by default.

Qualitative results are shown in Fig. 8. Visually, this reveals that the model detected the centers precisely. Specifically, the last row of the Fig. 8 highlights how small is the error between the ground truth and the prediction, as shown in the Table 2. Globally, the experimental result shows that our approach, in combination with some fine-tuning, is efficient to tackle the center detection task accurately while having a good generalization.

4.4.2. Contour segmentation

In this section, we present the results of the network performance and the quantitative metrics used for the contour segmentation task. Table 3 presents the performance of the model, both with and without data augmentation. As shown in the same table, we can see that the proposed network achieves good scores in both cases. We see a small improvement for the performance using random flip and random rotation. The performances of the segmentation network are reasonably good but the data augmentation has only a very limited impact on them.

Table 3
Model comparison with different parameters, the metrics being computed on the validation fold of the fir tree dataset.

Architecture	Augmentation	mean Dice/F1 score	Mean IoU
U-Net	/	0.875	0.780
U-Net	Flip and Rotate	0.878	0.785

Qualitative results are shown in Fig. 9. Visually, the network detected the masks of the wood log area precisely.

Taking into account that our model was only trained on fir trees, and to better highlight its robustness, we experimented it on many species with different shapes and sizes, none of which was seen in training. Fig. 10 presents the qualitative results of the robustness of our model. We noticed that our model was able to generalize the prediction with other species that have never been observed on training process. Unfortunately, due to the lack of labels for all the other species at the time of this writing, we cannot compute the metrics on all the dataset as we did for the centerline prediction.

This experimental result reveals that the U-Net architecture, with a good pre-processing combined with a good set of parameters and hyperparameters, is sufficient to capture the complex texture of tree slice CT images while having a good performance and robustness. The acacia and fir trees presented in Fig. 10 prove the ability of the model to generalize the prediction to different shapes and sizes, as the acacia tree is smaller than the example data used during

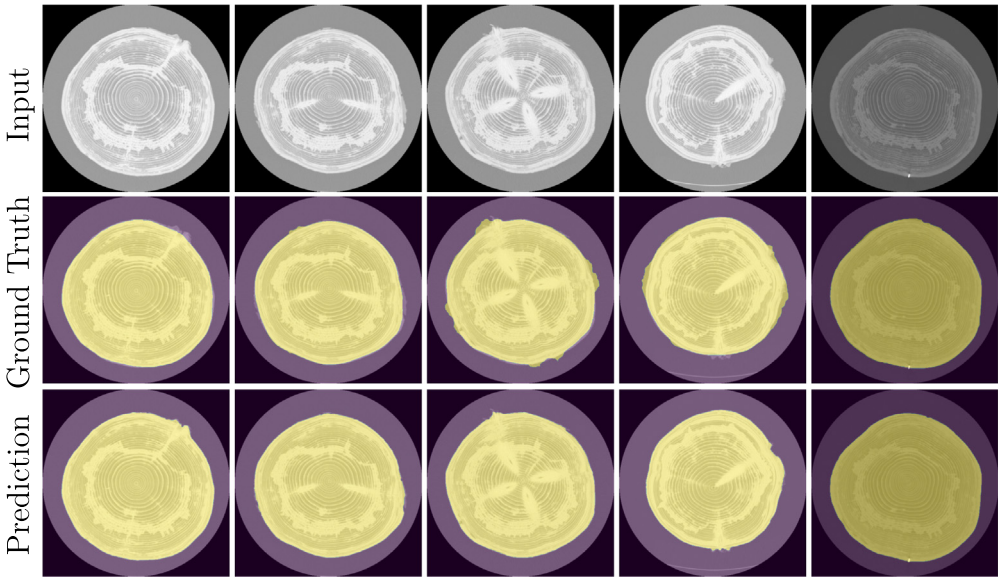


Fig. 9. Qualitative analysis of our model for different Fir tree species. The first row corresponds to the input images, the second row is the associated ground truth and the final one is the predictions. These samples all belong to the validation set.

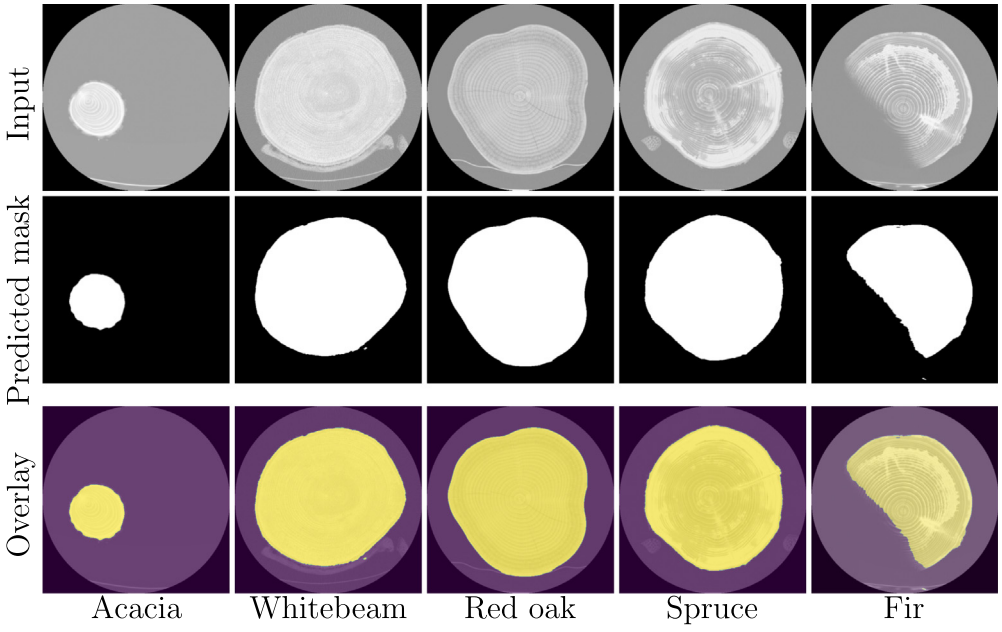


Fig. 10. Qualitative analysis of the robustness of our model with different tree species. The first row corresponds to the input images, the second row is the predictions, and the last one illustrates an overlay of the prediction.

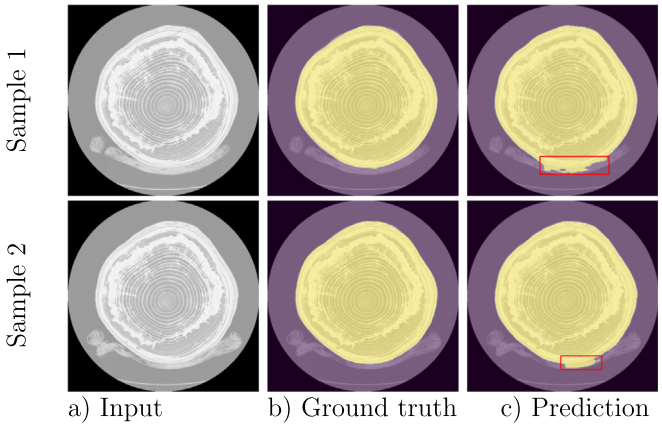


Fig. 11. Example of network mispredictions. The first column corresponds to the input images, the second column is the associated ground truth and the final one is the predictions. The red frame highlights the misprediction part.

Table 4
Model comparison with different parameters.

Architecture	Augmentation	Mean Dice/F1 score
U-Net	/	0.645
U-Net	Flipping & rotation	0.698

the training, and this fir tree log has a particular shape. Despite this, the model was able to segment precisely the entire visible part of the tree. More samples of different species are illustrated in appendix Appendix A.

Additionally, to better highlight the limits of our approach, we selected the validation samples for which the Dice score was the lowest. These samples, along with their predictions and ground truths, are shown in Fig. 11. On these samples, the difficulty seems to stem from the similarity between the trunk and the objects around them. Moreover, the model struggles to distinguish between the boundary of the trunk and the wedge that is very close to it. The misprediction is highlighted in a red frame in Fig. 11.

4.4.3. Knot segmentation

Finally, we experimented our proposed methodology for the knot segmentation task. Quantitative results of the best model minimizing the Dice score on the validation are shown in Table 4, with or without data augmentation. Quantitatively, on a per-case basis, we can see that the proposed architecture achieves a good score. We can also see that a simple data augmentation pipeline brings improvement, which validates the importance of variability of the dataset and the efficiency

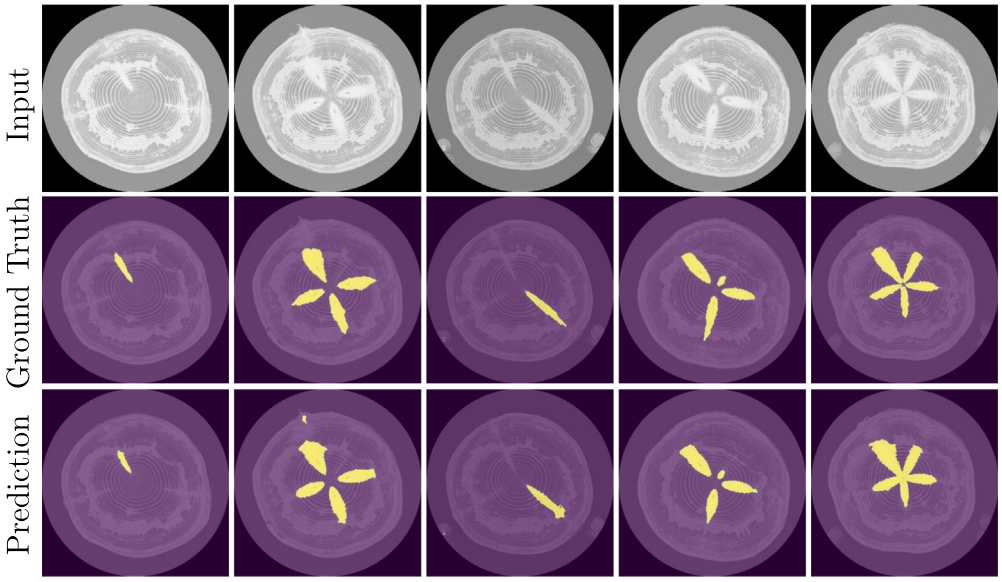


Fig. 12. Qualitative analysis of our model for different fir trees. The first row corresponds to the input images, the second row is the associated ground truth and the final one is the predictions.

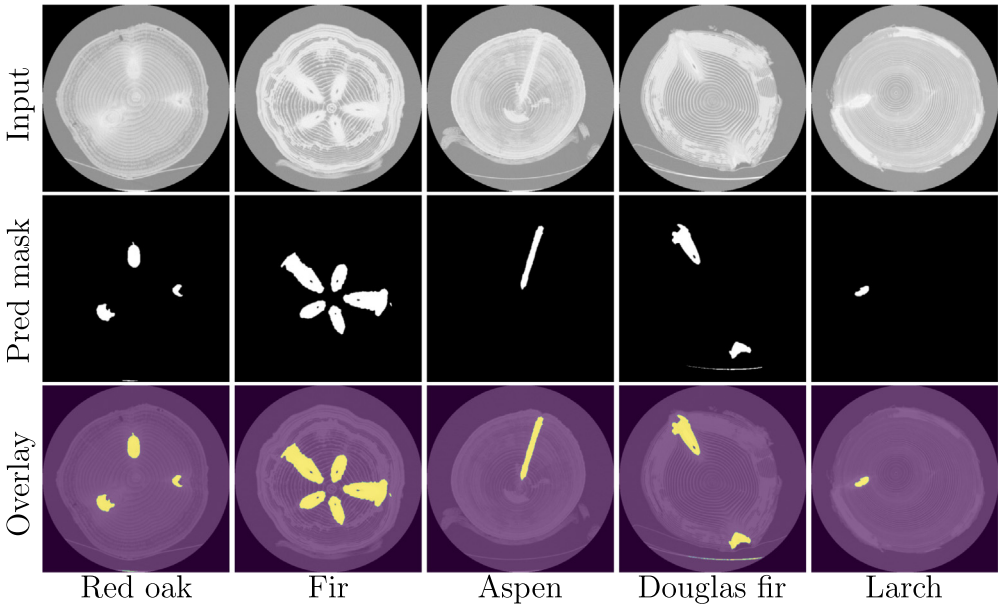


Fig. 13. Qualitative analysis of the robustness of the knot segmentation model with different tree species. The first row corresponds to the input images, the second row is the mask predictions, and the last one illustrates an overlay of the prediction.

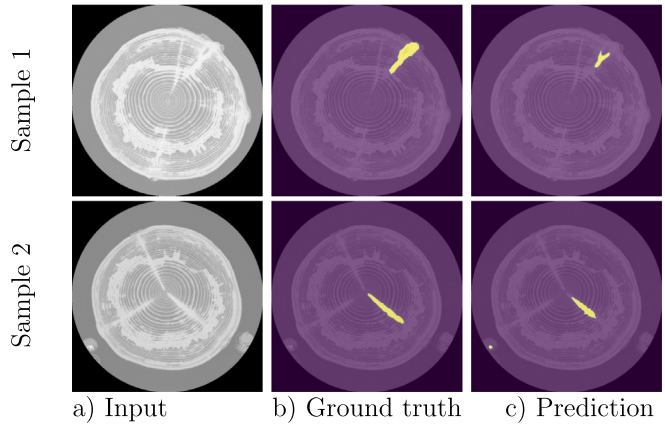


Fig. 14. Example of network prediction limits, the first column corresponds to the input image, the second one is the associated ground truth and the third one is the prediction on which we see that part of the knot is not correctly detected.

of using data augmentation on the performance of the segmentation network.

Qualitative results are shown in Fig. 12, we can see visually that our network detects the knots precisely, we also noticed that the proposed methodology exhibits a good performance on the segmentation task despite the limited annotated data (2504 images), which prove the efficiency of our approach.

Similarly to the contour segmentation task, and to better evaluate the robustness of our model, we tested it on different species that have never been observed during the training process. The Fig. 13 presents the qualitative results of the robustness, we noticed that the model was able to generalize well despite the challenge of the different structures and shapes of the knots. Unfortunately, due to lack of labels for all the other species at the time of this writing, we cannot compute the metrics on all the dataset as we did for the centerline prediction. The appendix Appendix B illustrates more samples for different species.

To better highlight the limits of our model on this challenging task, we selected a validation sample for which the Dice score was low. The sample, along with his prediction and ground truth, is shown in Fig. 14. On this sample, the difficulty seems to stem from the small size of the knot and the similarity between the density of the knot and the surrounding wood, which may explain why the network fails to predict the entire knot. As a side note, on this figure, one can notice some high-density radial features not labeled as knots in the ground-truth. These wood defects are not labeled as knots because they are not related to the growth of a branch. The network correctly ignores them in this segmentation task.

5. Conclusion

In this paper, we introduced an effective methodology based on deep convolutional neural networks to perform detection and prediction of tree semantic features in X-ray images. The proposed methods include three end-to-end pipelines that perform respectively the tree centerline regression, the contours and knots segmentations. The different results obtained have demonstrated the efficiency of this methodology and mainly its robustness on new unseen samples, which supports the relevance of deep learning based approaches for these tasks. These results also highlighted the generalization capacity of the models on different species with various shapes and sizes, despite the limited number of annotated data.

However, as of today, we also identified the following limitations. The first limitation comes from the small size of challenging details that the model was unable to capture on some species. This is especially true with the contours task as shown in Fig. 11 where the model struggle to distinguish the boundary of the trunk from the supporting wedge, and with the knot segmentation task (See Fig. 14) where the model fails to detect the entire knot in a challenging image characterized by the small size of the knot. The second limitation comes from the complexity of the knot structures and the density similarity with the surrounding wood in some instances, which leads to complete detection failure (Fig. 14). These limitations could be partially addressed by annotating more data with complex structures so as to help the model to capture small details and species-specific features. There are also various network architectural opportunities that could allow improving the performances of the presented approach. Compared to the single frame approach which uses only spatial information, we intend to explore the use of image sequences to take advantage of the sequential information as well. In addition, we are also considering adding a spatial attention block to focus on the most important, possibly distant, features to compute the predictions (Woo et al., 2018).

CRediT authorship contribution statement

Salim Khazem: Conceptualization, Methodology, Software, Formal analysis, Investigation, Writing – original draft. **Antoine Richard:** Writing – review & editing, Software, Data curation. **Jeremy Fix:** Supervision, Writing – review & editing, Software, Validation. **Cédric Pradalier:** Supervision, Writing – review & editing, Data curation, Validation.

Declaration of competing interest

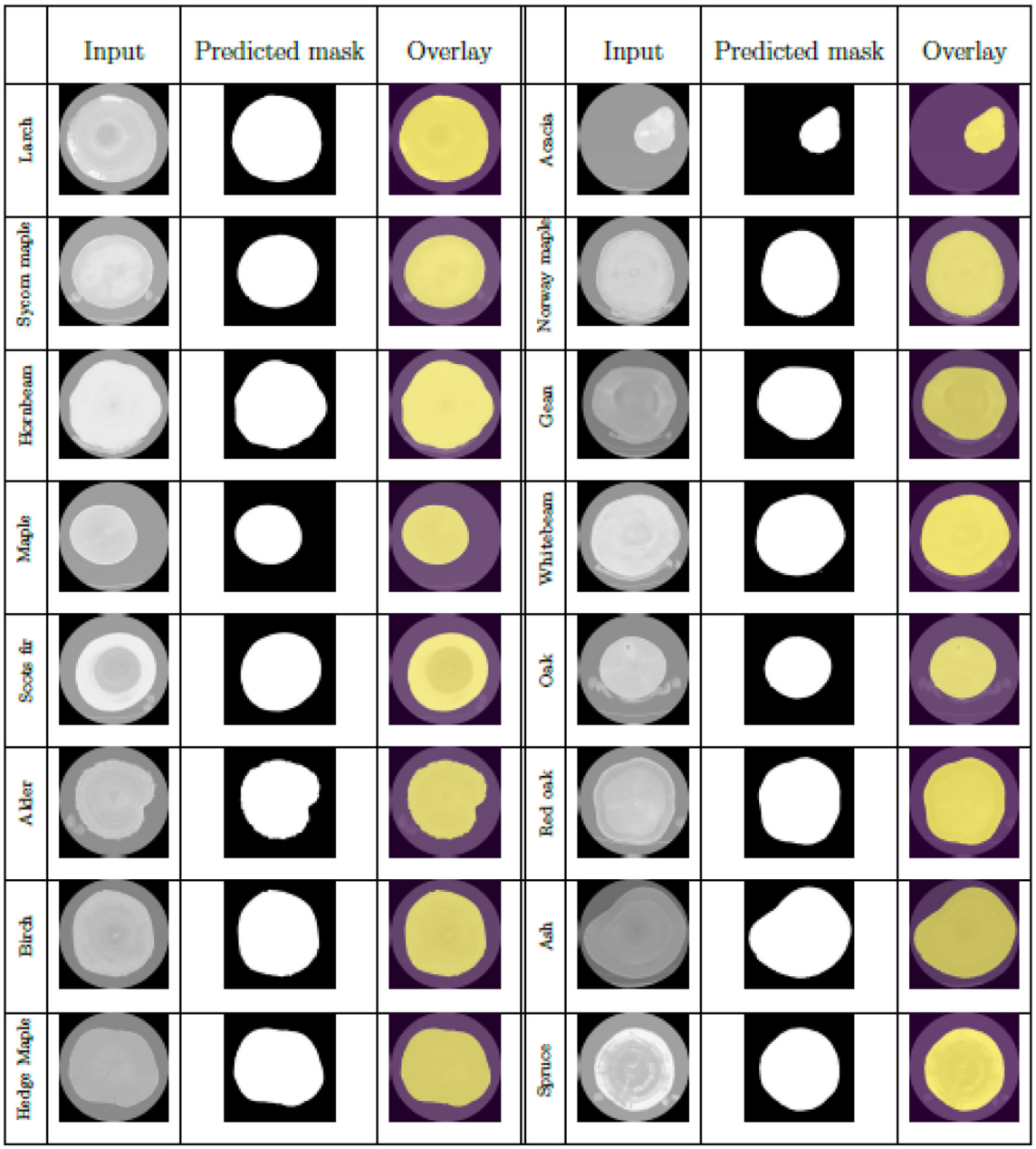
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This research was made possible with the support from the French National Research Agency, in the framework of the project WoodSeer, ANR-19-CE10-011.

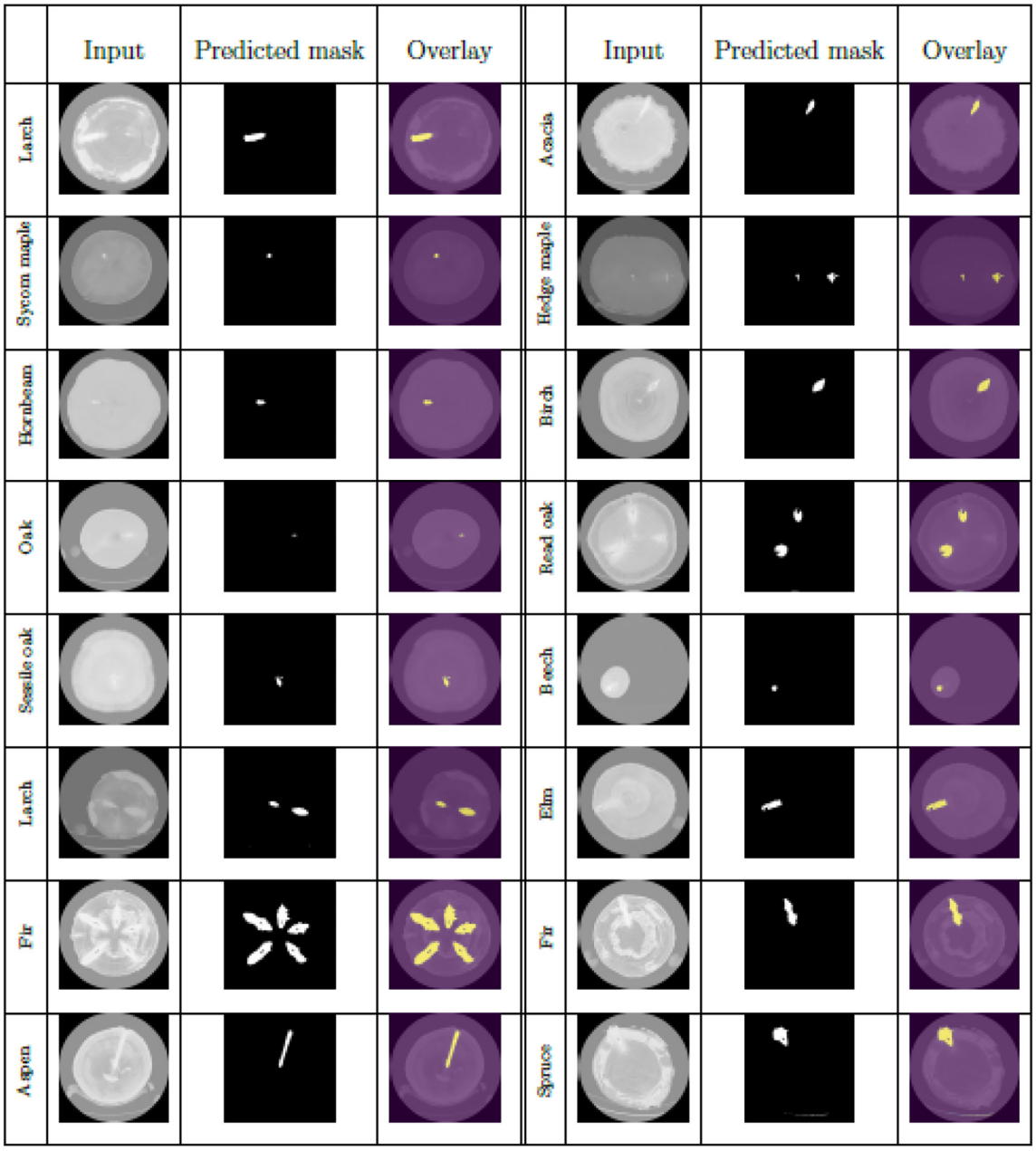
Appendix A. Contour segmentation

The figure below depicts a number of qualitative results of contour segmentation applied to various tree species not seen in the training dataset and for which the ground truth is not available.



Appendix B. Knots segmentation

The figure below depicts a number of qualitative results of knot segmentation applied to various tree species not seen in the training dataset and for which the ground truth is not available.



Appendix C. UNet architecture

Table C.5
Unet architecture. Conv(n) denotes a 2D convolutional layer with n kernels of size (3,3). Every convolution has a stride 1 and zero padding of size 1. MaxPool is a 2D max pooling layer with kernel size (2,2), stride 2.

Input (512×512)	
E_1:	Conv(64), BatchNorm, Relu
E_n: for n in {2.0.5}: with $c_0 = 128$ $4 \times \begin{bmatrix} \text{MaxPool} \\ \text{Conv}\ (2^{(n-2)} \times c_0), \text{BatchNorm, Relu} \\ \text{Conv}\ (2^{(n-2)} \times c_0), \text{BatchNorm, Relu} \end{bmatrix}$	
D_1:	ConvTranspose(512) Concat(E4) Conv(512), BatchNorm, Relu
D_2:	ConvTranspose(256) Concat(E3) Conv(256), BatchNorm, Relu
D_3:	ConvTranspose(128) Concat(E2) Conv(128), BatchNorm, Relu
D_4:	ConvTranspose(64) Concat(E1) Conv(64), BatchNorm, Relu
Output: Output shape (512×512)	Conv(n_{cls}), Sigmoid

References

Badrinarayanan, V., Kendall, A., Cipolla, R., 2016. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. arXiv:1511.00561 [cs] URL http://arxiv.org/abs/1511.00561.

Bhandarkar, S.M., Faust, T.D., Tang, M., 1999. CATALOG: a system for detection and rendering of internal log defects using computer tomography. Mach. Vis. Appl. 11, 171–190. https://doi.org/10.1007/s001380050100.

Buslaev, A., Iglovikov, V.I., Khvedchenya, E., Parinov, A., Druzhinin, M., Kalinin, A.A., 2020. Albumentations: fast and flexible image augmentations. Inf. 11, 125. https://doi.org/10.3390/info11020125.

Chaurasia, A., Culurciello, E., 2017. LinkNet: exploiting encoder representations for efficient semantic segmentation. IEEE Visual Communications and Image Processing (VCIP), pp. 1–4 URL.

Cortes, C., Vapnik, V., 1995. Support-vector networks. Mach. Learn. 20, 273–297.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N., 2021. An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale. 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3–7, 2021. OpenReview.net URL https://openreview.net/forum?id=YicbFdNTTy.

Gao, M., Song, P., Wang, F., Liu, J., Mandelis, A., Qi, D., 2021b. A novel deep convolutional neural network based on resnet-18 and transfer learning for detection of wood knot defects. J. Sensor. https://doi.org/10.1155/2021/4428964.

Gao, M., Chen, J., Mu, H., Qi, D., 2021a. A transfer residual neural network based on resnet-34 for detection of wood knot defects. Forests 12, 1–16. https://doi.org/10.3390/F12020212.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. Proc. IEEE Conf. Comput. Vis. Pattern Recognit, pp. 770–778.

He, K., Gkioxari, G., Dollár, P., Girshick, R., 2017. Mask r-cnn. Proceedings of the Ieee International Conference on Computer Vision, pp. 2961–2969.

Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H., 2017. Mobilenets: Efficient Convolutional Neural Networks for Mobile Vision Applications arXiv preprint arXiv:1704.04861.

Johansson, E., Johansson, D., Skog, J., Fredriksson, M., 2013. Automated knot detection for high speed computed tomography on pinus sylvestris l. and picea abies (l.) karst. Using ellipse fitting in concentric surfaces. Comput. Electron. Agric. 96, 238–245 URL.

Kerautret, B., Lachaud, J.O., 2009. Curvature estimation along noisy digital contours by approximate global optimization. Pattern Recogn. 42, 2265–2278. https://doi.org/10.1016/J.PATCOG.2008.11.013.

Kingma, D.P., Ba, J., 2015. Adam: a method for stochastic optimization. ICLR (Poster).

Krähenbühl, A., Kerautret, B., Debled-Rennesson, I., Longuetaud, F., Mothe, F., 2012. Knot detection in X-ray CT images of wood. International Symposium on Visual Computing. Springer, pp. 209–218.

Krähenbühl, A., Kerautret, B., Debled-Rennesson, I., 2013a. Knot segmentation in noisy 3d images of wood. International Conference on Discrete Geometry for Computer Imagery. Springer, pp. 383–394.

Krähenbühl, A., Kerautret, B., Debled-Rennesson, I., 2013b. Tkdetection: a Software to Detect and Segment Wood Knots. Imagen-a 3. URL https://hal.archives-ouvertes.fr/hal-01265531.

Krähenbühl, A., Kerautret, B., Debled-Rennesson, I., Mothe, F., Longuetaud, F., 2014. Knot segmentation in 3d ct images of wet wood. Pattern Recogn. 47, 3852–3869.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436–444. https://doi.org/10.1038/nature14539.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B., 2021. Swin transformer: hierarchical vision transformer using shifted windows. 2IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021. IEEE, pp. 9992–10002 https://doi.org/10.1109/ICCV48922.2021.00986.

Liu, Z., Mao, H., Wu, C., Feichtenhofer, C., Darrell, T., Xie, S., 2022. A Convnet for the 2020s. CoRR abs/2201.03545. URL https://arxiv.org/abs/2201.03545. arXiv:2201.03545.

Longo, B., Brüchert, F., Becker, G., Sauter, U., 2019. Validation of a ct knot detection algorithm on fresh Douglas-fir (pseudotsuga menziesii (mirb.) franco) logs. Ann. For. Sci. 76. https://doi.org/10.1007/s13595-019-0812-4.

Lopes, D.J.V., dos Santos Bobadilha, G., Grebner, K.M., 2020. A fast and robust artificial intelligence technique for wood knot detection. BioResources 15, 9351–9361. https://doi.org/10.15376/BIORES.15.4.9351-9361.

Marcos, D., Tuia, D., Kellenberger, B., Zhang, L., Bai, M., Liao, R., Urtasun, R., 2018. Learning deep structured active contours end-to-end. Proc. IEEE Conf. Comput. Vis. Pattern Recognit, pp. 8877–8885.

Micikevicius, P., Narang, S., Alben, J., Diamos, G., Elsen, E., Garcia, D., Ginsburg, B., Houston, M., Kuchaiev, O., Venkatesh, G., et al., 2017. Mixed Precision Training arXiv preprint arXiv:1710.03740.

Micikevicius, P., Narang, S., Alben, J., Diamos, G., Elsen, E., Garcia, D., Ginsburg, B., Houston, M., Kuchaiev, O., Venkatesh, G., Wu, H., 2018. Mixed precision training. arXiv: 1710.03740 [cs, stat] URL http://arxiv.org/abs/1710.03740.

Mustra, M., Delac, K., Grgic, M., 2008. Overview of the dicom standard. 50th international symposium ELMAR. IEEE, pp. 39–44.

Norlander, R., Grahn, J., Maki, A., 2015. Wooden knot detection using convnet transfer learning. Scandinavian Conference on Image Analysis. Springer, pp. 263–274.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al., 2019. Pytorch: an imperative style, high-performance deep learning library. Adv. Neural Inf. Proces. Syst. 32.

Perez, L., Wang, J., 2017. The Effectiveness of Data Augmentation in Image Classification Using Deep Learning arXiv preprint arXiv:1712.04621.

Redmon, J., Farhadi, A., 2018. Yolov3: An Incremental Improvement arXiv preprint arXiv: 1804.02767.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. arXiv:1505.04597 [cs] URL http://arxiv.org/abs/1505.04597.

Shorten, C., Khoshgoftaar, T.M., 2019. A survey on image data augmentation for deep learning. J. Big Data 6, 60. https://doi.org/10.1186/s40537-019-0197-0.

Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition. In: Bengio, Y., LeCun, Y. (Eds.), 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings URL http://arxiv.org/abs/1409.1556.

Sudre, C.H., Li, W., Vercauteren, T., Ourselin, S., Jorge Cardoso, M., 2017. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support. Springer, pp. 240–248.

Tan, M., Le, Q., 2019. Efficientnet: rethinking model scaling for convolutional neural networks. International Conference on Machine Learning. PMLR, pp. 6105–6114.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H., 2021. Training data-efficient image transformers & distillation through attention. In: Meila, M., Zhang, T. (Eds.), Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18–24 July 2021, Virtual Event. PMLR , pp. 10347–10357 URL http://proceedings.mlr.press/v139/touvron21a.html.

Woo, S., Park, J., Lee, J.Y., Kweon, I.S., 2018. Cbam: convolutional block attention module. Proceedings of the European Conference on Computer Vision (ECCV), pp. 3–19.

Xiong, Y., Liao, R., Zhao, H., Hu, R., Bai, M., Yumer, E., Urtasun, R., 2019. Upsnet: a unified panoptic segmentation network. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8818–8826.