Localisation of Defects in Volumetric Computed Tomography Scans of Valuable Wood Logs

Davide Boscaini[1(✉)], Fabio Poiesi[1], Stefano Messelodi[1], Ayman Younes[2], and Donato A. Grande[2]

[1] Fondazione Bruno Kessler, Povo, TN, Italy
{dboscaini,poiesi,messelod}@fbk.eu
[2] Meccanica del Sarca S.p.A., Dro, TN, Italy
{a.younes,d.grande}@sarca.it

Abstract. We present a novel pipeline to efficiently localise defects in volumetric Computed Tomography (CT) scans of valuable wood logs. We couple a 2D detector applied independently on each scan slice with a multi-object tracking approach processing detections along the scan direction to localise the defects in 3D. Our solution is designed to meet the real-time requirements of modern production lines, to optimise the wood sawing operations for high-quality final products and to reduce wood waste as well as carbon footprints. We effectively embedded our defect localisation algorithm in the Meccanica del Sarca S.p.A.'s production pipeline achieving a reduction of their economic loss by 7% compared to the previous years.

1 Introduction

Defects in precious wood logs are one of the most important cause of big economic loss in wood industries, producing also unnecessary carbon footprints. Meccanica del Sarca S.p.A., an Italian company that deals with the mechanical process of precious wood logs, estimated that they lost about 405K Euro in 2017, 430 K Euro in 2018, and 424 K Euro in 2019 due to wood waste. Usually, the log production process starts with an operator that inspects the external surface of a log searching for, or predicting, internal defects. On the one hand, the inspection of the log surface may not provide enough evidence to suggest the presence of internal defects. On the other hand, a visible defect on the log surface may suggest to discard the whole wood log despite being undamaged internally. The two main factors that lead to the economic loss are the incorrect classification of defected logs (i.e. material waste) and the late identification of defects during the production process (i.e. time waste). Therefore, the need for a automated system to accurately detect defects in the early stage of the log production process is key. If a defect is detected in the early production stage, it is possible to optimally plan the sawing operations in order to avoid defects and to minimise waste.

Wood log interiors can also be analysed using Computed Tomography (CT) [7] (Fig. 1). CT produces a set of image slices that are typically used by

A. Del Bimbo et al. (Eds.): ICPR 2020 Workshops, LNCS 12664, pp. 692–704, 2021.
https://doi.org/10.1007/978-3-030-68799-1_50

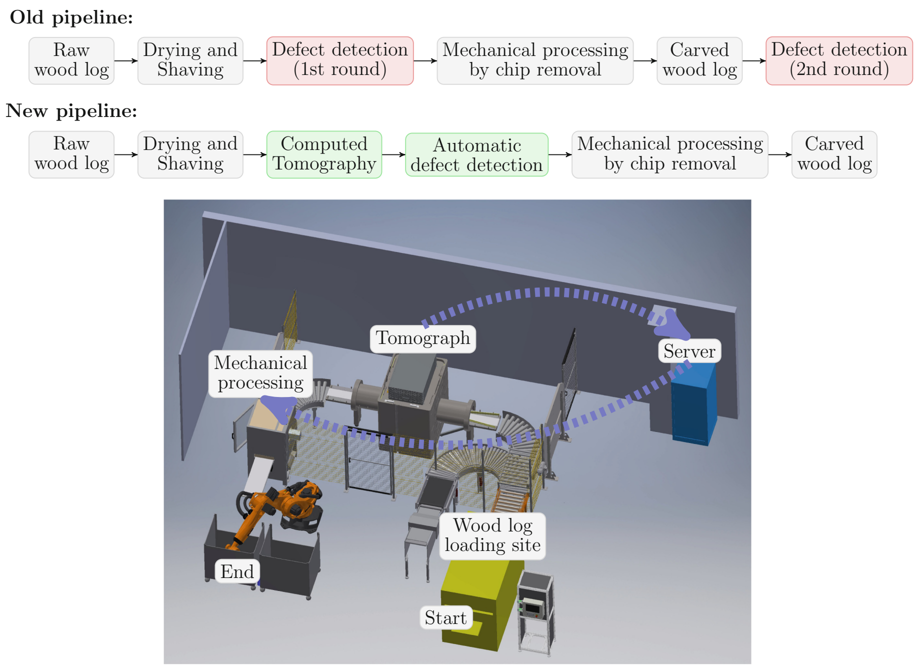


Fig. 1. Current wood log processing pipeline. Top row: previous Meccanica del Sarca S.p.A.'s pipeline where defects were found manually by experts that visually inspected the external surface of the wood logs before and after their mechanical processing. Bottom row: current wood processing pipeline where internal parts of the logs are acquired through Computed Tomography and the defect detection is performed automatically using our deep-learning based algorithm. Data flow is represented by blue arrows. (Color figure online)

operators to verify the presence of defects. This job is laborious and in certain circumstances not well defined. There is wide scope for subjective interpretation of what a defect is, which may lead to annotation mistakes. Elements of ambiguity may be related to the management of adjacent defects. There can be regions containing several small and spread defects, that for some experts it could be regarded as a single defect. The concept of proximity, considering the difficulty in defining defect boundaries, can be subjective.

The problem of wood defect detection has already been tackled in literature [1,3,11,13–15]. Proposed methods used algorithms based on Artificial Neural Networks [11], edge-based image analysis [1], Support Vector Machine classification applied to colour features [3] or, more recently, based on deep-learning, i.e. Faster-RCNN [13]. Because the CT scan of a wood log produces its 3D reconstruction, 3D detection methods could in principle be used to detect its internal defects [4]. However, we decided not to use 3D object detectors for two reasons. The operator would need to wait the end of the CT scan to examine the log 3D reconstruction, thus affecting the real-time requirements of the production lines. The defect may have an irregular shape, therefore, because the output of a 3D detector typically is an axis-aligned 3D bounding box, this may include a significant percentage of good material. Differently, we propose to deal with the problem of defect detection by coupling a 2D object detector operating on each scan slice with a multi-object tracking approach along the scan direction. The 2D object detector is modelled as a end-to-end deep neural network that takes the CT slices as input and generates the bounding boxes coordinates and categories of likely defects as output. The multi-object tracking acts as a refinement post-processing to linearly interpolate bounding boxes in the case of miss-detections and to prune outlier detections. We perform tracking using a tracking-by-detection algorithm formulated as bipartite graph matching.

2 Our Approach

Our defect localisation pipeline is composed of a module that detects defects as 2D bounding boxes for each scan slice, and of a module that tracks these bounding boxes along the scan direction using current and past information only.

2.1 2D Defect Detection

We base our detection module on the Single Shot MultiBox Detector (SSD) that was proposed for object detection in RGB images [8]. Figure 2 depicts a block diagram of the proposed architecture. We denote our model with Φ_Θ, where Θ is the collection of its learnable parameters. Φ_Θ is composed of three main modules: the feature extraction core, the classification head and the regression head. The main difference between Φ_Θ and the original SSD model [8] is in the classification and regression heads: we use less skip connections (two instead of six) and we place them at shallower levels (54×54 feature map locations instead of 38×38, and 27×27 instead of 19×19, respectively). These modifications are motivated by the fact that the defects we are aiming to detect are, on average, much smaller than the typical object size the original SSD model was developed for. Anticipating the skip connections allow us to extract feature maps at a higher resolution grid.

Ψ_Ω is the feature extraction module with Ω its learnable parameters. Ψ_Ω takes a mini-batch of b CT images of size 440×440 in input and outputs features at two depth levels, $\mathbf{h}^1 = \Psi^1_{\Omega^1}(\mathbf{x})$ of size $(b, 512, 54, 54)$ and $\mathbf{h}^2 = \Psi^2_{\Omega^2}(\mathbf{h}^1)$ of size $(b, 1024, 27, 27)$, where $\Psi_\Omega = \Psi^2_{\Omega^2} \circ \Psi^1_{\Omega^1}$. The precision of the input CT images is set to 16-bit in order not to lose possibly relevant information. The early layers of Ψ_Ω implements a VGG-16 backbone [12] truncated before the classification layer and endowed with Batch Normalization [5], and are followed by three custom layers: MaxPool2d, Conv2d(512, 1024) + ReLU, and Conv2d(1024, 1024) + ReLU.

We associate a default bounding boxes of different aspect ratios, called anchors, to each feature map cell of $\mathbf{h}^1, \mathbf{h}^2$. For each anchor the classification

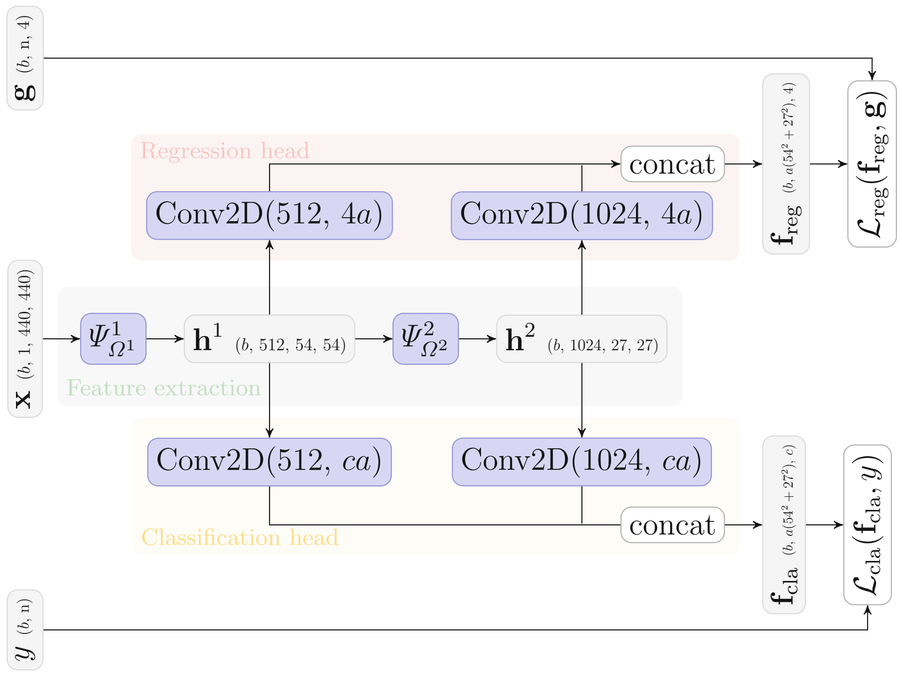


Fig. 2. Block diagram of our SSD model applied to CT scans. Colour key. Grey: input/output tensors. Blue: parametric layer. White: non-parametric layer. (Color figure online)

head predicts c scores that indicates the presence of a class instance inside it. The regression head predicts the offsets to apply to the 4 anchor coordinates to fit the ground-truth bounding box.

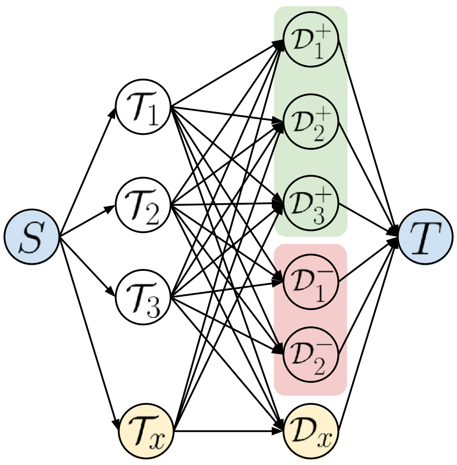
A classification loss $\mathcal{L}_{\text{cla}}$ measures the consensus between the output of the classification head $\mathbf{f}_{\text{cla}}$ and the ground-truth class y. Similarly, a regression loss $\mathcal{L}_{\text{reg}}$ measures the error between the output of the regression head $\mathbf{f}_{\text{reg}}$ and the offsets between the anchors and the ground-truth bounding boxes $\mathbf{g}$. Without loss of generality, we configure our detector for binary classification: a bounding-box can contain either a defect or not. We use the cross entropy loss as classification loss,

$$\mathcal{L}_{\text{cla}}(\mathbf{f}_{\text{cla}}, k) = -\log \frac{\exp \mathbf{f}_{\text{cla}}(k)}{\sum_k \exp \mathbf{f}_{\text{cla}}(k)},$$

and the smooth L1 loss, also known as Huber loss [2], as regression loss,

$$\mathcal{L}_{\text{reg}}(\mathbf{f}_{\text{reg}}, \mathbf{g}) = \text{SmoothL1}(\mathbf{f}_{\text{reg}}, \mathbf{g}).$$

The Huber loss is defined as a squared L2 norm if the absolute error falls below 1 and as an L1 norm otherwise. During training we optimise a linear combination of the two losses, i.e. $\mathcal{L} = \mathcal{L}_{\text{cla}} + \lambda \mathcal{L}_{\text{reg}}$. In our experiments we set $\lambda = 0.1$ by cross-validation.



Edges	Cost	Capacity
$e(S, \mathcal{T}_i)$	0	1
$e(S, \mathcal{T}_x)$	0	$N_{\mathcal{D}}$
$e(\mathcal{T}_i, \mathcal{D}_j^+)$	$\phi(\mathcal{T}_i, \mathcal{D}_j)$	1
$e(\mathcal{T}_i, \mathcal{D}_j^-)$	$1.1 \cdot \phi(\mathcal{T}_i, \mathcal{D}_j)$	1
$e(\mathcal{T}_i, \mathcal{D}_x)$	ν_s	1
$e(\mathcal{T}_x, \mathcal{D}_j)$	ν_n	1
$e(\mathcal{T}_x, \mathcal{D}_x)$	0	$N_{\mathcal{D}}$
$e(\mathcal{D}_j, T)$	0	1
$e(\mathcal{D}_x, T)$	0	$N_{\mathcal{T}}$

Fig. 3. Graph matching formulation to solve our data association (tracking) problem. Nodes in this graph include source (S), sink (T), detections ($\mathcal{D}_j$), track states ($\mathcal{T}_i$), and two proxy nodes, i.e. $\mathcal{T}_x$ and $\mathcal{D}_x$, to allow track initialisation and track termination, respectively. $\mathcal{D}_j^+$ is a strong detection, $\mathcal{D}_j^-$ is a weak detections and $\mathcal{D}_j$ is a generic detection. $\phi(\mathcal{T}_i, \mathcal{D}_j)$ is the association cost function between $\mathcal{T}_i$ and $\mathcal{D}_j$.

2.2 Multi-defect Tracking

We filter out and associate detections across slices using online tracking. Given a set of $N_{\mathcal{D}}$ detections computed on the current slice we aim to associate them to the set of $N_{\mathcal{T}}$ tracks that are computed on the previous slice. Let $\mathcal{T}_i$ be the ith track and $\mathcal{D}_j$ be the jth detection. We formulate the tracking problem as a bipartite graph matching and solve it using Minimum Cost Flow [9]. We define our graph as $\mathcal{G} = (N, E)$, where N represents the set of nodes and E the set of edges. Tracking states and detections are the nodes of $\mathcal{G}$. Nodes in this graph include source (S), sink (T), detections ($\mathcal{D}_j$), track states ($\mathcal{T}_i$), and two proxy nodes, i.e. $\mathcal{T}_x$ and $\mathcal{D}_x$, to allow track initialisation and track termination, respectively. Specifically, our tracking algorithm initialises, associates and terminates tracks based on costs. Initialisation and termination costs are hyper-parameters set by us. The association cost is a function that depends on the position and on the bounding-box size between a detection and its predicted track state. Let $\phi(\mathcal{T}_i, \mathcal{D}_j)$ be the association cost function between $\mathcal{T}_i$ and $\mathcal{D}_j$. $\phi(\cdot)$ is a linear combination of distance between the centres, widths and heights of last bounding box of $\mathcal{T}_i$ and the bounding box of $\mathcal{D}_j$. We denote $e(n_1, n_2)$ as the edge between node n_1 and node n_2. Each edge is characterised by the cost and a capacity. Figure 3 shows the graph formulation of our tracking algorithm.

The solution of this graph is a one-to-one association between track states and detections. In our graph formulation we embed the weak and strong detection tracking model proposed in [10]. Strong detections are detections with confidence above the threshold δ^+, weak detections are detections with confidence between

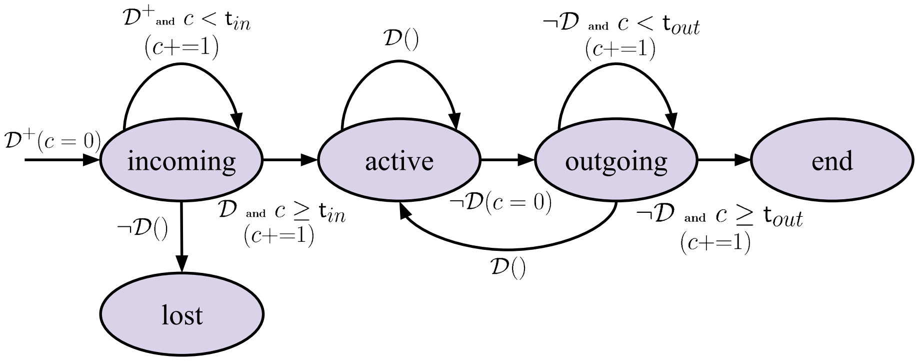

$\mathcal{D}^+$ and $c < \mathsf{t}_{in}$ (c+=1)
$\mathcal{D}()$
$\neg\mathcal{D}$ and $c < \mathsf{t}_{out}$ (c+=1)
$\mathcal{D}^+(c=0)$
incoming
active
outgoing
end
$\neg\mathcal{D}()$
$\mathcal{D}$ and $c \geq \mathsf{t}_{in}$ (c+=1)
$\neg\mathcal{D}(c=0)$
$\neg\mathcal{D}$ and $c \geq \mathsf{t}_{out}$ (c+=1)
$\mathcal{D}()$
lost

Fig. 4. Track state management. Incoming, lost, active, outgoing and end are the possible states of each track. A new track always starts in the incoming state. Key. input: $\mathcal{D}$ ($\neg\mathcal{D}$) detection is (not) associated to the track. Edge label: input [**and** condition] (action). c: counter.

the threshold δ^- and δ^+. These thresholds should be set such that $\delta^+ > \delta^-$. Strong detections are used for track initialisation and for tracking, whereas weak detections are used for tracking only. Let $\mathcal{D}_j^+$ be a strong detection, $\mathcal{D}_j^-$ be a weak detection and $\mathcal{D}_j$ be a generic detection. For each slice processed by the detector we solve the graph matching problem. For each track we build a state machine to manage its evolution. Figure 4 shows our state machine. An unassociated strong detection triggers the initialisation of a new track, which begins from the incoming state. If this track is successfully associated to a number of t_{in} consecutive strong detections, it goes in the active state. Subsequent associations in the active state use both strong and weak detections. In the case of a miss-detection, i.e. an unsuccessful association, the track goes in the outgoing state. If the association fails for more than t_{out} consecutive slices the track ends. Otherwise if the track is associated to a new detection in subsequent slices before reaching t_{out}, the track state goes back to the active state and the slices where miss-detections occurred are filled with linearly interpolated bounding boxes. The linear interpolation uses the information from the associated bounding boxes.

3 Experimental Results

3.1 Data Acquisition and Normalization

CT scans of the wood logs were captured at the Meccanica del Sarca S.p.A.'s premises using a MiTO tomograph developed by Microtec s.r.l., the world leading wood scanning solutions provider. During the CT acquisition a wood log is transported by a conveyor belt moving at a constant velocity. An X-ray scanner moves with a spiral pattern opposite to the conveyor belt direction to scan the log (Fig. 5, left). A spin of the X-ray scanner produces a 2D slice of the 3D wood

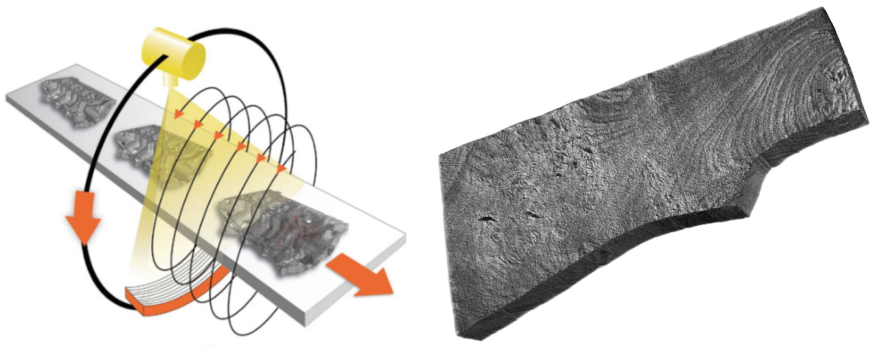


Fig. 5. Computed Tomography (CT) acquisition process and a wood log 3D reconstruction from its CT scans.

log. Multiple 2D image slices are captured while the logs moves on the conveyor belt. The collection of all slices form the 3D structure of the wood log (Fig. 5, right). The tomograph acquisition accuracy is set to 0.5 mm for both the 2D and 3D scan directions. This means that a voxel of the 3D reconstruction of the wood log corresponds to a cube of material of size 0.5 mm.

Slices are stored as signed 16-bit single-channel 2D images. Each pixel of the image contains an intensity value that is expected to be correlated with the material density. However, Fig. 6 shows that in practice the histogram of the intensities of the pixels of a CT scan is bimodal. The first mode corresponds to air (intensity approx. 0) plus noise related to acquisition artifacts. The second mode corresponds to the actual wood material. To focus our analysis on the wood material we normalise each image between 200 and 1200.

3.2 Datasets

We collected a dataset of 175 CT scans of wood logs of various sizes. Each scan contains between 412 to 1144 slices, for a total of 149,237 CT images. Each CT scan is composed of about 1000 slices, where each slice may contain up to tens of defects. We split the data in training and testing sets with proportions of 85% and 15%, respectively. The training set contains 150 CT scans, i.e. 127,954 CT images. The test set contains 25 CT scans, i.e. 21,283 CT images.

Experts from Meccanica del Sarca S.p.A. carefully annotated the wood defects by drawing axis-aligned bounding boxes for each slice. The annotation process was particularly difficult, time consuming and prone to errors. The difficulty was due to the fact that experts were used to identify defects by inspecting the surface of the wood log. They had to acquire some experience to understand how to identify defects from the CT slices. To speed-up this annotation process, instead of asking the experts to annotate every slice we proposed to annotate every 5–10 slices and then to linearly interpolate the missing bounding boxes. We discovered that there is wide scope for subjective interpretation of what a

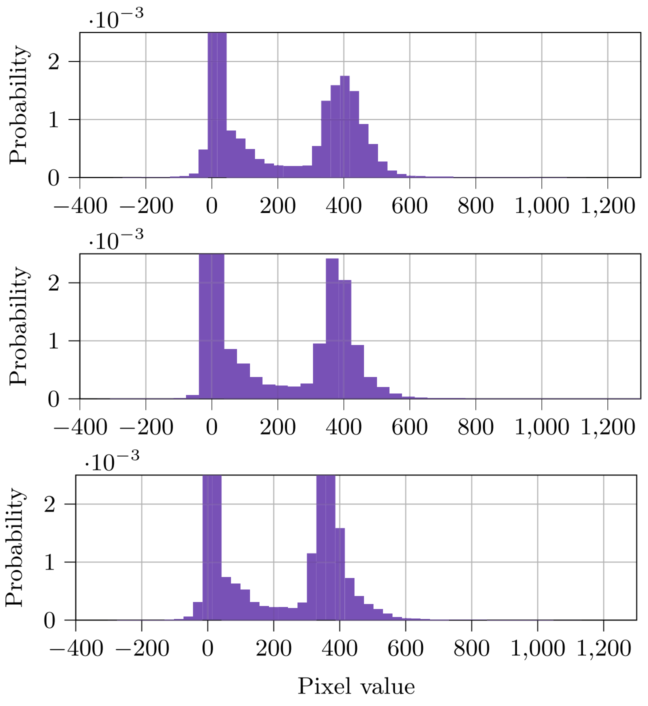


Fig. 6. Histograms of three CT scans that show the distribution of intensity values. The left-hand side mode represents empty regions (i.e. air), while the right-hand side mode represent the region with material.

defect is, which often led to annotation mistakes. To reduce these mistakes we had several iterations with the experts from Meccanica del Sarca S.p.A. to review and correct the annotated bounding boxes. We experimentally confirmed that accurate annotations lead to a considerable improvement in the performance of our deep-learning based defect detection algorithm.

3.3 Experimental Setup

Training. We train our detector for 200 epochs using the Stochastic Gradient Descent (SGD) as optimiser. We set SGD to an initial learning rate of $1e-2$. We decrease the learning rate by a factor of 0.75 every 20 epochs. We use a weight decay with a factor of $1e-03$ for regularisation. We train on an NVidia GeForce GTX 1080 with 8 GB RAM.

Inference. At inference time we apply Non-Maximum Suppression (NMS) [8]. NMS filters out bounding-box predictions that have a confidence lower than t_{conf} and that have an overlap bigger than t_{NMS}. We choose $t_{\text{conf}} = 0.5$ (or 0.05) and $t_{\text{NMS}} = 0.5$ in our experiments.

Pretraining. We use a pretrained version of VGG-16 layers on the ImageNet dataset for the feature extraction module. The other layers are trained from scratch. Although ImageNet is a RGB-image dataset, which is quite different from our CT scan dataset, we found that this pretraining enables us to achieve better performances in fewer epochs.

Data Augmentation. We use data augmentation to avoid overfitting on the training set. Specifically, we used (i) horizontal and vertical flipping of the image, each with a probability of 0.5 to be applied to the input image, (ii) random cropping also with probability set to 0.5, followed by a resizing of the crop region to the resolution of the input image, (iii) data normalisation using the mean μ and standard deviation σ of training data, i.e. $\mu \approx 0.074, \sigma \approx 0.012$. As far as random cropping is concerned, let h, w be the height and width of the input image, and $\bar{h}, \bar{w}$ be the height and width of the crop region, and $\bar{y}, \bar{x}$ be the top-left coordinates of the crop region. In our experiments, we set $\bar{h} \sim \mathcal{U}(0.3h, h), \bar{w} \sim \mathcal{U}(0.3w, w)$, $\tilde{y} \sim \mathcal{U}(100, h - \bar{h})$, and $\tilde{x} \sim \mathcal{U}(50, w - \bar{w})$. If the cropped region has an aspect ratio smaller than 0.5 or bigger than 2 the hyper-parameters described above are re-sampled until this condition is met. The bounding boxes whose centres are outside the cropped region are discarded.

Detector Configuration. We set the number of anchors to $a = 9$. The anchors cover different aspect ratios, obtained by multiplying the width and height of the feature map cell by the following factors: $(1, 1)$, $(2, 2)$, $(3, 3)$, $(2, 1)$, $(1, 2)$, $(4, 2)$, $(2, 4)$, $(6, 3)$, and $(3, 6)$.

Tracking Configuration. The tracking hyper-parameters are set through cross-validation using the validation set: $\mathtt{t}_{in} = 0$, $\mathtt{t}_{out} = 3$, $\delta^{+} = 0.5$, $\delta^{-} = 0.05$.

3.4 Analysis of the Results

We quantify the localisation performance using Precision, Recall and F1 Score [6]. We assess the quality of our pipeline with and without the tracking module activated, and by performing an ablation study on the key parameters.

Table 1 shows the results obtained using $t_{\text{eval}} = 0.25$, i.e. a bounding box is considered correctly estimated if its overlap with the ground-truth bounding box is greater than 1/4 of its area. The first row of Table 1 shows the detection performance using bounding boxes predicted with a confidence greater than 0.5. Although the results show that we can achieve a high precision with this configuration (Prec. $\approx$ 90%), the number of missed defects is high (Rec $\approx$ 55%). This results in a F1-score of 68.4. The second row of Table 1 shows the detection results when the confidence threshold is lowered to 0.05. As expected, this affects the precision (Prec $\approx$ 40%), whereas the recall score increases to $\approx$ 80%. Figure 7 shows some examples of detection results. The latter is a suitable case to postprocess with tracking given its ability in filtering out false positives and to interpolate miss-detections. The third row of Table 1 shows how tracking effectively improves Precision, despite worsening Recall. We empirically observed that there are several situations where detections with a confidence above δ^{+}

Table 1. Quantitative comparison results in terms of True Positives (TP), False Negatives (FN), False Positives (FP), Precision (Prec.), Recall (Rec.) and F1-score (F1).

Method	TP	FN	FP	Prec	Rec	F1
SSD ($t_{\text{conf}} = 0.5$)	2475	1971	312	88.8	55.7	68.4
SSD ($t_{\text{conf}} = 0.05$)	3724	722	5832	39.0	**83.8**	53.2
Our ($t_{\text{conf}} = 0.05$)	3203	1243	1449	**68.9**	72.0	**70.4**

Table 2. Ablation on the data augmentation. Hyper-parameters used: $\lambda = 1\text{e}-01$, learning rate = $1\text{e}-03$, weight decay = $1\text{e}-04$, $t_{\text{conf}} = 0.5$, $t_{\text{eval}} = 0.5$. Results reported are best values achieved on the test set.

Data augm.	TP	FN	FP	Prec	Rec	F1
data norm.	2075	2371	986	67.8	46.7	55.3
data norm. + crop	2018	2428	514	**79.7**	45.4	57.8
data norm. + crop + flip	2291	2155	999	69.6	**51.5**	**59.2**

are not enough consistent over consecutive slices to become tracks, thus deemed false positives and filtered out by tracking. Figure 8 shows examples of tracking results where false-positive detections are prunned and miss-detections are corrected.

Table 2 reports the ablation study on different data transformation configurations that we used for the training of our detector. We obtained the best performance in terms of F1-score by combining data normalisation with random cropping and image flipping. We found that data augmentation is key to avoid overfitting. Table 3 reports the ablation study using different tracking parameters. We can observe that a high value of the threshold to decide between strong

Table 3. Ablation on the tracking hyper-parameters. First row show the upper bound we can reach in recall.

t_{in}	t_{out}	δ^+	δ^-	TP	FN	FP	Prec	Rec	F1
0	3	0.05	0.05	3792	654	7142	34.7	85.3	49.3
0	3	0.1	0.05	3700	746	4608	44.5	83.2	58.0
0	5	0.1	0.05	3721	725	5029	42.5	83.7	56.4
1	5	0.1	0.05	3692	754	4303	46.2	82.0	59.1
0	3	0.4	0.05	3328	1118	1820	64.7	74.9	69.4
0	3	0.5	0.05	3203	1243	1449	68.9	72.0	**70.4**
1	5	0.5	0.05	3202	1244	1453	68.8	72.0	**70.4**
0	3	0.6	0.05	3020	1426	1260	70.6	67.9	69.2
0	3	0.7	0.05	2932	1514	1061	73.4	66.0	69.5

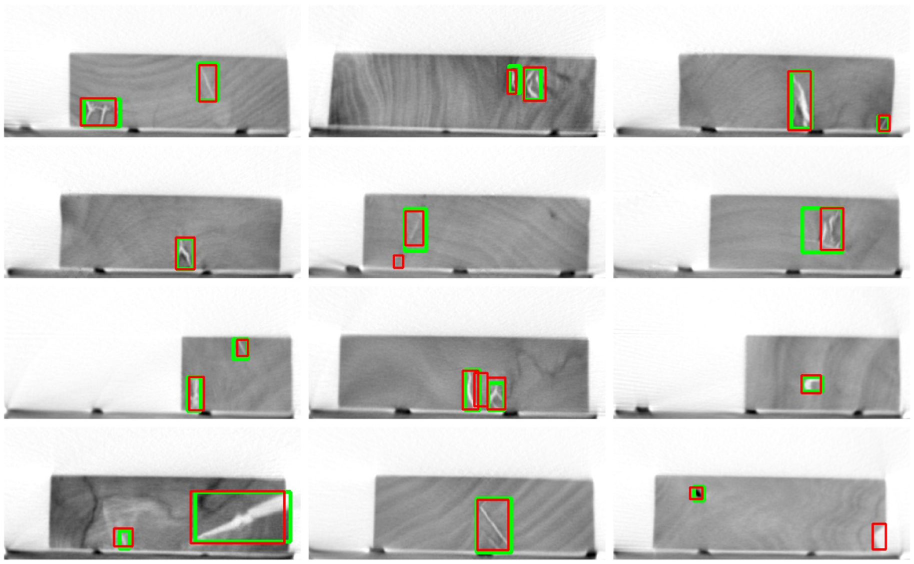


Fig. 7. Examples of detection results. Our detector can handle defects at different scales. Some detected regions are difficult to judge whether they are a defected, e.g. second row - second and third figure. There are then extreme cases where regions at the border of the log are detected as defects. Bounding-box key. Green: ground-truth. Red: estimated detection.

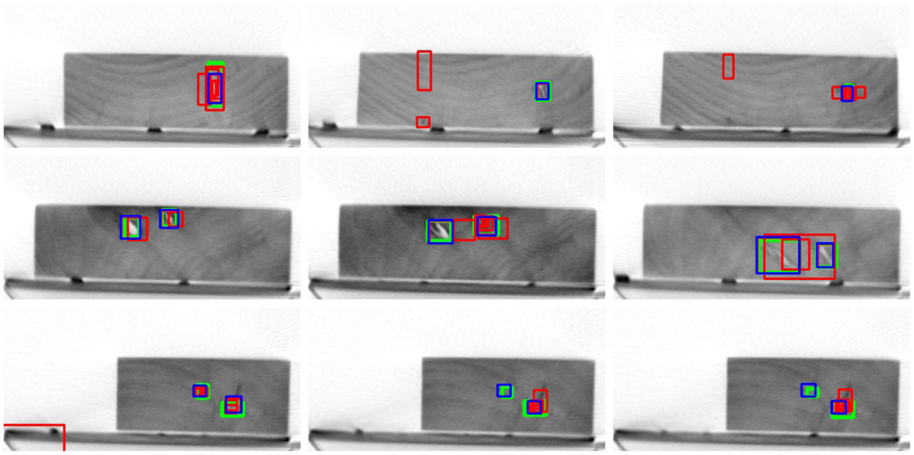


Fig. 8. Examples of tracking results. The first and second rows show how tracking can effectively filter out spurious detections. The last row shows the ability of tracking in interpolating miss-detections (top-left bounding boxes). Bounding-box key. Green: ground-truth. Red: estimated detection. Blue: estimated tracking state. (Color figure online)

and weak detections have a positive effect on the Precision, but that it affects the Recall. Based on the application at hand, a user can tune these parameters to increase or decrease the sensitivity of the approach in order achieve the desired output quality.

4 Conclusions

We presented a deep-learning based algorithm to localise defects in volumetric Computed Tomography scanned wood logs. We showed how to perform 3D defect localisation via 2D object detection and multi-object tracking in order to meet the real-time requirements of the production line. We trained our models on annotations made by experts that deal with the wood production industry. Although the annotation process seems straightforward for deep-learning engineers, we experienced several challenges in instructing the experts. Annotation accuracy is critical to deploy reliable data-driven algorithms on a real production lines. We experimentally showed that we achieved promising localisation performance, which especially contributed to reduce the company's economic loss by 7% compared to the previous years. This experience helped us understand that greater effort must be put into the creation of intuitive mechanisms for data annotation and into comprehensive protocols to localise well-defined defects.

Acknowledgments. This work has been developed within a collaboration between FBK and Meccanica del Sarca S.p.A. and funded by the "Programma operativo FESR 2014–2020 della Provincia di Trento".

References

1. Bhandarkar, S., Luo, X., Daniels, R., Tollner, E.: Detection of cracks in computer tomography images of logs. Pattern Recogn. Lett. **26**, 2282–2294 (2005)
2. Girshick, R.: Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2015)
3. Gu, I.Y.H., Andersson, H., Vicen, R.: Wood defect classification based on image analysis and support vector machines. Wood Sci. Technol. **44**, 693–704 (2010). https://doi.org/10.1007/s00226-009-0287-9
4. Gwak, J., Choy, C., Savarese, S.: Generative Sparse Detection Networks for 3D Single-shot Object Detection. arXiv:2006.12356, June 2020
5. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: Proeedings of the International Conference on Machine Learning (ICML) (2015)
6. ISO 5725–1: Accuracy (trueness and precision) of measurement methods and results, Part 1: General principles and definitions, International Organization for Standardization (1994)
7. Kak, A.C., Slaney, M.: Principles of computerized tomographic imaging. IEEE Press, the Institute of Electrical and Electronics Engineers Inc., New York (1999)
8. Liu, W., et al.: SSD: single shot MultiBox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016, Part I. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2

9. Mehlhorn, K.: Algorithms on Graphs. Data Structures and Algorithms. Graph Algorithms and NP-Completeness. Monographs in Theoretical Computer Science. An EATCS Series, vol. 2. Springer, Heidelberg (1984). https://doi.org/10.1007/978-3-642-69897-2
10. Sanchez-Matilla, R., Poiesi, F., Cavallaro, A.: Online multi-target tracking with strong and weak detections. In: Hua, G., Jégou, H. (eds.) ECCV 2016, Part II. LNCS, vol. 9914, pp. 84–99. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48881-3_7
11. Sarigul, E., Abbott, A., Schmoldt, D.: Progress in analysis of computed tomography (CT) images of hardwood logs for defect detection. In: Proceedings of the Tenth International Conference on Scanning Technology and Process Optimization in the Wood Industry (2003)
12. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: Proceedings International Conference on Learning Representations (ICLR) (2015)
13. Urbonas, A., Raudonis, V., Maskeliunas, R., Damasevicius, R.: Automated identification of wood veneer surface defects using faster region-based convolutional neural network with data augmentation and transfer learning. Appl. Sci. **22**, 4898 (2019)
14. Yuhan, Q., Zhou, Y., Xu, J., Ge, Z.: Development of a wood computed tomography imaging system using a butterworth filtered back-projection algorithm. For. Prod. J. **68**, 147–156 (2018)
15. Zhao, P., Wang, C.K.: Hardwood Species Classification with Hyperspectral Microscopic Images. J. Spectro. (2019)