Smart Agricultural Technology 4 (2023) 100201

Contents lists available at ScienceDirect

Smart Agricultural Technology

journal homepage: www.journals.elsevier.com/smart-agricultural-technology

Tree log identification using convolutional neural networks

Eero Holmström [a,*], Antti Raatevaara [a,b,c], Jonne Pohjankukka [a], Heikki Korpunen [d], Jori Uusitalo [b]

[a] Natural Resources Institute Finland (Luke), Latokartanonkaari 9, Helsinki 00790, Finland
[b] Department of Forest Sciences, University of Helsinki, Latokartanonkaari 7, 00014, Finland
[c] MiCROTEC Innovating Wood Oy, Klovinpellontie 1-3, Espoo 02180, Finland
[d] NIBIO, Høgskoleveien 8, Ås 1433, Norway

ARTICLE INFO

Edited By Editor: Stephen Symons

Keywords:
Deep learning
Wood procurement
Traceability
Scots pine

ABSTRACT

The identification of individual tree logs along the wood procurement chain is a coveted goal within the forest industry. The tracing of logs from the sawmill back to the forest would support the legal and sustainable sourcing of wood, as well as increase the resource efficiency and value of harvested timber. In this work, using a dataset of thousands of Scots pine (*Pinus sylvestris* L.) log end images displaying varying perspectives, lighting, and aging effects, we develop and assess log identification methods based on deep convolutional neural networks. The estimated rank-1 accuracy of our final model on an independent test set of 99 logs is 84 and 91% when allowing for random rotations of the log ends and when keeping each log at approximately fixed orientation, respectively. We estimate the scaling of these methods up to a template pool size of 493 logs, which reveals a weak dependence of accuracy on pool size for logs at fixed orientation. The deep learning approach gives superior results to a classical local binary pattern method, and appears feasible in practice, assuming that pre-filtering of the log database can be leveraged depending on the use case and properties of the queried log image. We make our dataset publicly available.

1. Introduction

The mechanical wood industry converts harvested wood into boards, planks, furniture, beams, and other long-lasting wood products. This industry constitutes a significant sector of forestry in the Nordic countries, in particular Sweden and Finland [1]. However, wood procurement for Nordic sawmills, at the heart of the mechanical wood industry, is subject to several challenges.

First, cut-to-length harvester machines are operated using incomplete information regarding the internal quality of trees. This leads to undesired combinations of wood quality and dimensions being bucked in the forest, which decreases the value of sawn lumber. Furthermore, imprecise bucking in the presence of defects such as rot [2] and sweep [3] leads to resource loss in harvested wood. Furthermore, sawlogs stored at the sawmill yard are at peril of deteriorating when stored for extended periods of time during warm seasons. In addition, illegal logging is a significant problem globally, contributing to biodiversity loss, soil erosion, and lost state revenue [4].

All of these issues could be alleviated by the identification of individual sawlogs throughout the wood procurement chain. To begin with, the tracing of individual logs from the sawmill back to the forest would allow connecting measurements at the sawmill, such as X-ray and 3D optical scanning of logs, to imaging data on the same logs as collected by the harvester [5–7]. This would make it possible to learn mappings between freshly cut log end images and wood properties, which would imply higher value and resource efficiency for lumber through bucking optimization [5,3]. In addition, the identification of an individual log would permit finding the elapsed time since the tree from which the log originates was felled in the forest. This would help to mitigate resource loss at sawmills due to wood deteriorating from fungi and insects when kept in storage prior to sawing for excessive periods. Moreover, tracing logs to their location of harvest could be used to enforce the legal and responsible sourcing of wood in cases where the origin of incoming sawlogs is not known through other mechanisms.

In addition, pairing sawmill measurements with the precise geographical origin of the standing tree from which a log originates would propagate the uncovering of connections between detailed harvest site characteristics [8–10], external properties of standing trees

[*] Corresponding author.
E-mail address: eero.holmstrom@luke.fi (E. Holmström).

https://doi.org/10.1016/j.atech.2023.100201
Received 16 February 2023; Accepted 16 February 2023
Available online 18 February 2023

[11–14], and wood quality for optimizing sawmill production planning. Such models could also be used in the development of various precision forestry applications.

Current log identification methods are based on RFID tags, painted labels, chemical tracers, etc. [15,16], but these are either expensive or cumbersome or can only account for logs at the batch level. Imaging is an alternative approach to log identification, which benefits from requiring no physical marking of logs. Classical approaches, where features are extracted and analyzed using handcrafted algorithms have been presented for log end RGB images [7]. However, the stability of these methods is questionable due to their design for specific circumstances and the large amount of parameters which need to be tuned via expert knowledge.

In contrast, deep learning approaches based on convolutional neural networks (CNN) display an excellent track record in biometric identification tasks [17–19] and might well provide a powerful tool for log identification [20]. Recent work on CNN-based identification of wood from images of bark appears promising [21,22], but considering the damage that bark undergoes during harvesting and transportation as well as in sorting at the sawmill, the log end should present a more stable attribute for identification.

In this work, we develop and assess methods based on deep CNNs for accomplishing the identification of individual logs using log end face images. For this purpose, we collect a dataset of thousands of Scots pine (*Pinus sylvestris* L.) log images with variations in imaging angle and lighting, as well as a time gap of five days between the probe and template images of each log. We then consider a broad range of deep CNN architectures to produce feature vectors which are discriminative enough for direct use in identifying individual logs via distance to candidate logs in embedding space. In addition, we implement the classical benchmark method of local binary pattern histograms [23,24] for comparing our deep learning methods against. Finally, we evaluate the deep learning and classical identification methods on independent test data, and discuss the scalability and practical issues of CNN-based log identification in real forest operations.

2. Materials and methods

2.1. Collecting the image data

The log identification process studied in this work consisted of comparing a given log end probe image against template images previously registered in a database, and finding the best match out of these. A schematic of a possible log identification system in a real cut-to-length forest operation utilizing such an identification process is illustrated in Fig. 1.

To develop our methods, we collected an original data set of images through photographing a total of 493 individual Scots pine logs at a clear-cut harvest site in South-East Finland in September 2021. We performed the photography in two sessions, each lasting one working day.

The first session took place a couple of days after the logs had been harvested. In this session, we photographed logs in order to simulate the situation where a log is registered by the harvester machine. We first photographed each log directly head-on, which we call a “headshot” picture, and then from a total of four different angles, to the left, right, above, and below the log face center, which we call “angled shots”. These images acquired in the first photography session we collectively call “fresh” images. During the first imaging session, we tagged each log with a piece of paper showing the unique number of the log, so that the exact same logs could be photographed in the second session.

The second photography session took place after a period of five days from the first session. We returned to the site and repeated the same procedure. The purpose of the second session was to simulate the situation where logs are photographed further up the logistics chain. We call the images captured in the second photography session “aged” images. The full photography scheme is illustrated in Fig. 2. The total amount of images captured was 4930.

We photographed the logs in their roadside storage piles. The vast majority of the imaged specimens were sawlogs, but we also photographed some logs of sawlog diameter which had been classified for pulp production due to defects such as sweep. A significant proportion of all photographed logs were butt logs. For each log, we targeted the butt

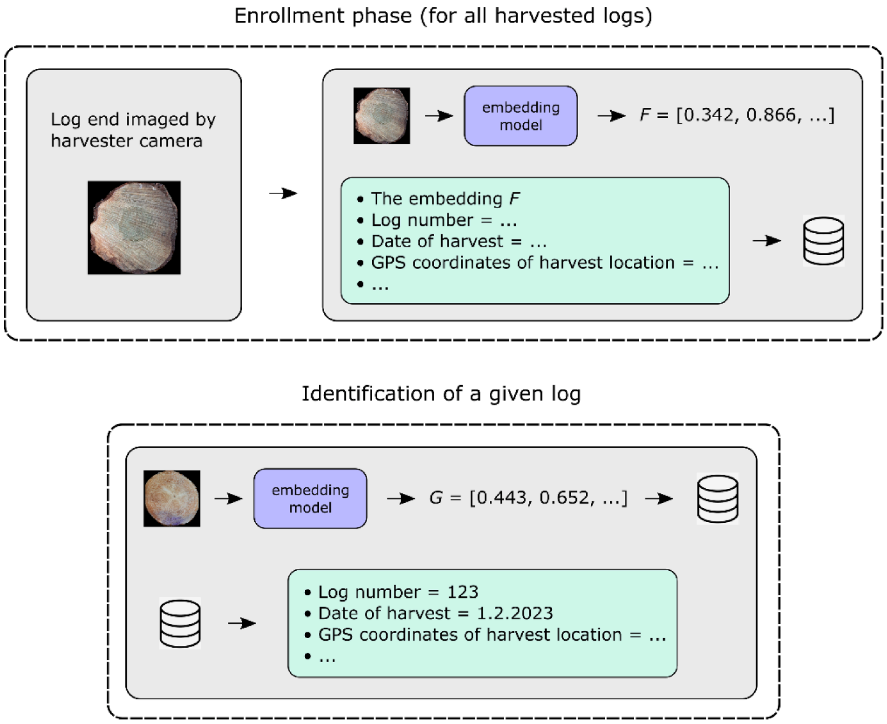
Enrollment phase (for all harvested logs)
Log end imaged by harvester camera
embedding model
F = [0.342, 0.866, ...]
• The embedding F
• Log number = ...
• Date of harvest = ...
• GPS coordinates of harvest location = ...
• ...
Identification of a given log
embedding model
G = [0.443, 0.652, ...]
• Log number = 123
• Date of harvest = 1.2.2023
• GPS coordinates of harvest location = ...
• ...

Fig. 1. Schematic of a possible system for identifying an individual log in cut-to-length harvesting. In the enrollment phase, each log created by the harvester machine is imaged. An embedding, i.e., feature vector, is then computed for each log and sent to a server to be registered in a database along with other useful information on the log. In the identification phase, the end face of a log to be identified is photographed at, e.g., a sawmill, producing a probe image. The embedding for the probe image is computed and sent to the server. The server then returns the information for the log with the most closely matching embedding vector.

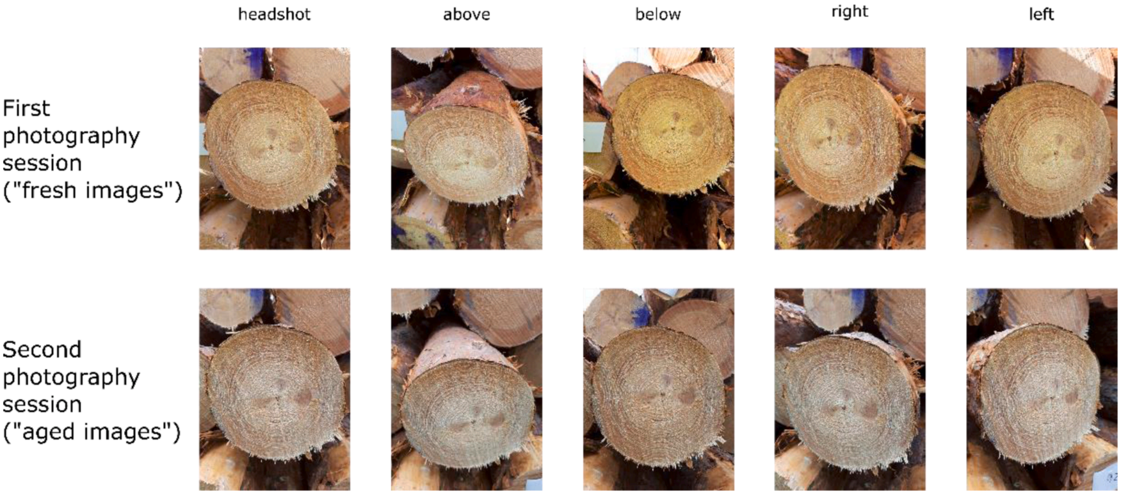
headshot
above
below
right
left
First photography session ("fresh images")
Second photography session ("aged images")

Fig. 2. Illustration of the photography scheme used in data collection. There was a time difference of five days between the first and second photography sessions. All images here depict the same log.

end. Some logs that we photographed may have originated in the same stem. However, due to there being a distance of several meters between any two such butt-end images along the stem, correlations between the images should not be significant enough to affect our results. The camera used for imaging was the 25 megapixel main camera of the Samsung Galaxy A50 smartphone on standard settings.

2.2. *Preparing the image data*

Because we photographed the logs in roadside piles, each image contained parts of neighboring logs around the targeted log (Fig. 2). Since both the fresh and aged image of a log were taken with the same surrounding logs in the image, this could have lead to the identification method leveraging the features of these surroundings instead of focusing on the actual object of interest, i.e., the specifically targeted log. To prevent this from happening, we segmented out each log end image along the edge of the log face and set the segmented log end onto a black background.

We performed the segmentation using a Detectron2 [25] Mask R-CNN implementation with a ResNet101 [26] backbone. We trained the Mask R-CNN instance segmentation model with 143 log end images and corresponding polygon annotations, which we drew using the LabelMe-tool [27]. Since each log had been imaged in a roadside pile and was therefore surrounded by other logs, we chose to use the mask instance that was positioned at the center of the image. The images produced were of a square aspect ratio. The trained segmentation model was able to predict masks with an average Jaccard similarity index of 0.9 when compared to hand-drawn ground-truth masks in a small, independent test sample of 50 log end images. In cases where the Mask R-CNN model failed to produce an acceptable outcome, we drew the masks manually. As assessed via visual inspection over the final set of 4930 log end images, the segmentation quality was very good in all but a handful of cases.

After the segmentation, we randomly split the total data set of 493 logs into a training set (296 logs), a development set (98 logs, from hereon referred to as the *dev set*), and a held-out test set (99 logs).

While a log often appeared similar in both the fresh and dried images, these two sets of images generally differed in appearance (Fig. 3). The lighting, including the position of the sun, caused changes in the shadowing on log ends as well as the recorded colors of the logs. The log face color also changed due to drying in some cases. In addition, the drying sometimes introduced cracks into the log, and in some cases, preexisting cracks grew in size during the five-day period. Contrasts between resinous or wet areas and the rest of the log face generally decreased when moving from fresh to aged images. In some cases, vegetation such as hay was visible in both fresh and aged images for a log, but in other cases the vegetation disappeared between the images. If a log end was soiled with dirt, more or less the same dirt was visible on both imaging days. Also, some of the images were out of focus, further contributing to variance between log images, not only between the fresh and aged images of a log. In addition, in some cases the bark was included in the segmented image, while in others it was not. The full dataset of 4930 segmented log images is publicly available at https://doi.org/10.5281/zenodo.7281916.

2.3. *Developing the log identification methods*

The specific goal of the identification process was the following: *Given an angled shot of a log taken on the second shooting day, find the headshot image of the same log taken on the first shooting day*. This setup simulated the situation where the log is originally registered into a database via its fresh headshot image in the forest at time of harvesting, and then during transport or at the sawmill, a new image of the log is acquired from a non-headshot angle, with the aim of identifying the log in question.

For a given model and dataset (train, dev, test), we computed the rank-1 identification accuracy as follows. We first registered all the first-day headshots of logs into the “database”, i.e., enrolled the templates. Then, we iterated over the four non-headshot images of all logs taken on the second shooting day, i.e., the probe images, and found the closest match for each of these in the database. Each probe image had exactly one correct counterpart in the database (Fig. 1). The rank-1 identification accuracy was equal to the number of correct matches divided by the number of all matching attempts, i.e., the number of probe images. Equivalently, the accuracy was equal to the mean over the individual identification results ($1 = $ success, $0 = $ fail) over the probe images. Using these individual identification results, we computed an estimate for the standard error of the (mean) accuracy.

We studied two scenarios, (1) one where the log rotation was approximately fixed over all images of the log, i.e., as captured in the original photographs, and (2) one where the log rotation was uniformly randomized over 360°. For scenario (1), we subjected images in the training set to small random rotations during training. For scenario (2) we subjected logs in the training set to random rotations over the full range of 0 to 360°, and in addition, all images of the dev and test sets were randomly and permanently rotated over the full range of 0 to 360°. Both scenarios are illustrated in Fig. 4.

We considered two different approaches to achieving log

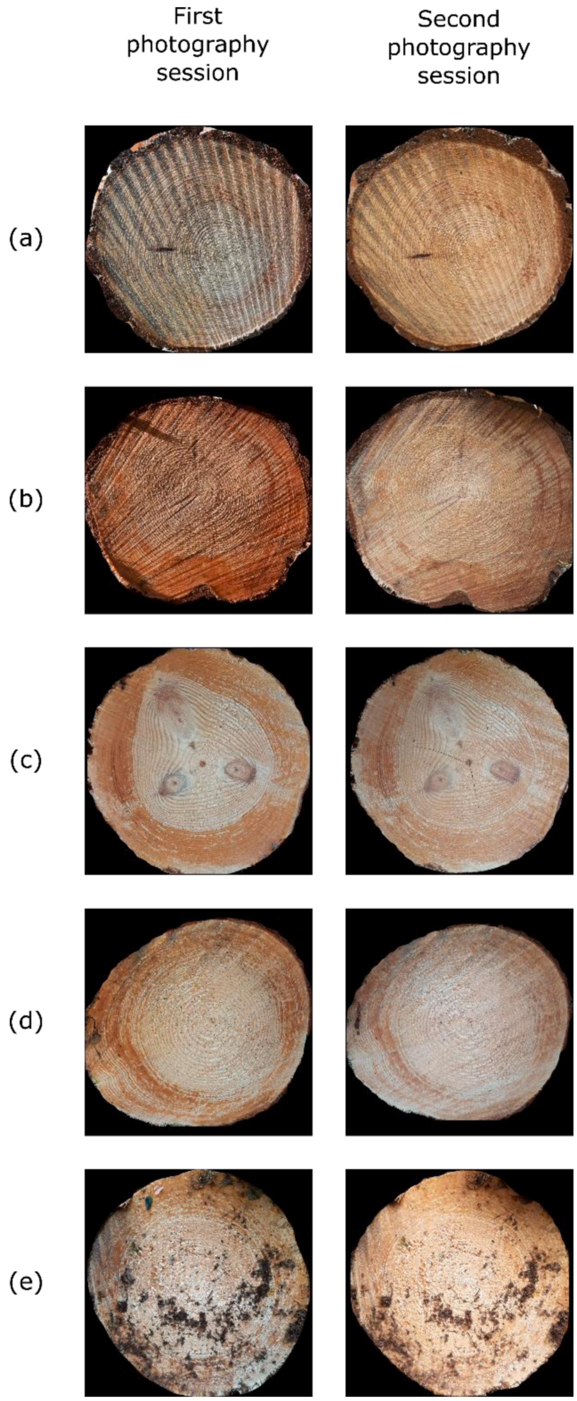

Fig. 3. Examples of how log appearance changed between the two imaging sessions. In order to highlight effects arising from other factors besides the imaging angle, all the shown images are headshots (see Section 2.1 for details). In many cases, changes in color and shadowing can be observed between the fresh and aged photograph of a log (a, b). Some logs developed cracks between the two photography sessions (c), while the contrast between resinous and non-resinous and/or moist and dry areas diminished in others (d). When the log end was soiled with dirt, practically the same dirt pattern was observed in both the fresh and aged images (e).

identification: (1) a deep-learning method based on CNNs, and (2) a local binary pattern histogram method to serve as a classical benchmark for the former. Approaches (1) and (2) were adapted for both identification scenarios, i.e., logs at fixed rotation and logs at random rotations. We describe the development of the deep learning approach in Sections 2.3.1, 2.3.2, 2.3.3, and 2.3.4. After this, we describe the development of the classical approach in Section 2.3.5.

2.3.1. *Outline of the deep learning approach*

In this approach, our aim was to use a deep CNN as a feature extractor for producing an embedding vector for a given log end image, which is robust and discriminative enough to allow accurately identifying the log among a pool of registered candidates in the varying imaging conditions represented by our dataset. We developed an end-to-end embedding model, which we first used to compute the deep feature vector for each fresh headshot of a log and register these into a database. Then, to identify a given log based on its aged, angled image, we computed the deep feature vector F for any such image and found the registered feature vector in the database with the shortest Euclidean distance to F. If the two images depicted the same log, the identification was considered successful. Otherwise, the identification was considered failed.

We used Keras [28] and TensorFlow [29] for all the deep learning work in this study. The embedding model structure consisted of a deep CNN in the role of a feature extractor followed by two fully connected layers of sizes N_{FC1} and N_{FC2}, respectively, and a final fully connected “embedding” layer of size N_{E}, which output an L2-normalized feature vector f, setting the image embedding onto an N-dimensional hypersphere (Fig. 5).

2.3.2. *Triplet loss training*

In analogy with face recognition methods for identifying humans, we harnessed triplet loss training [19,30] to optimize the network for the task at hand. To accomplish this, we constructed a triple-Siamese network which took as input three images, i.e., a triplet: the fresh headshot image of a log (the “anchor image” a), an aged angled shot of the same log (a “positive example” p), and an aged angled shot of some other log (a “negative example” n). The network then output two numbers: the squared Euclidean distance $|f_a - f_p|^2$between the feature vectors of a and p, and the same for the feature vectors of a and n. The training loss for a single triplet was computed as

$$\max\left(0,\ |f_a - f_p|^2 + \varepsilon - |f_a - f_n|^2\right) \tag{1}$$

Successfully training the network to minimize triplet loss makes the distance between the anchor and negative example larger than the distance between the anchor and positive example by a margin of at least ε, the so-called *triplet loss margin*. By training the network via the gradient of the mean triplet loss over each training batch with respect to model weights, the model learns to produce embeddings that give large distances between a and n and small distances between a and p. Selecting which triplets, out of all possible ones, to feed to the network during training can be crucial to obtaining a good model [19,30]. Here we adopted the following approach.

At the beginning of each epoch, we first found all the positive pairs a and p in the training set. After this, for each positive pair (a, p), we found all the negative examples n that extended the pair into a triplet which satisfied the condition

$$|f_a - f_n|^2 < |f_a - f_p|^2 + \varepsilon' \tag{2}$$

where $\varepsilon' = \varepsilon$. In other words, we looked for triplets where the embedding model at its current state incorrectly ordered the positive and negative examples in terms of distance from the anchor image, and which therefore produced a non-zero loss contribution. Once all the negative examples that qualify had been found for the pair (a, p), we uniformly randomly chose one of these to complete the triplet. In other words, we used *hard* and *semi-hard triplets* for the training.

However, once the training had advanced for some time, negative examples that fulfilled Eq. 2 could not always be found for a pair (a, p). When this happened, we increased the margin ε' for the triplet selection

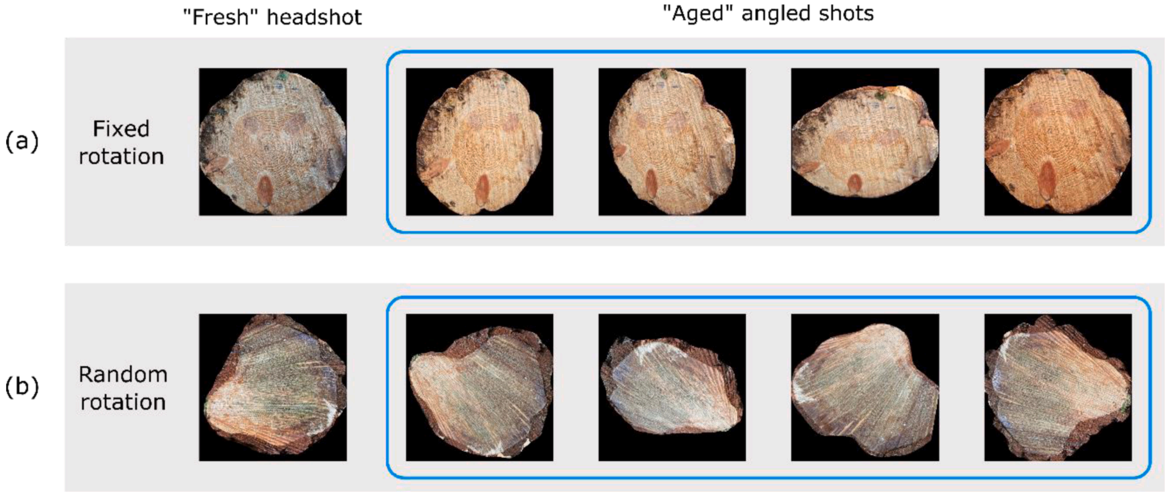

Fig. 4. Example of a first-day headshot and all the second-day angled shots for a log in both studied identification scenarios: (a) log rotation is approximately fixed (b) log rotation is uniformly random. The goal in the identification was to correctly match a given "aged" angled shot to the "fresh" headshot of the same log.

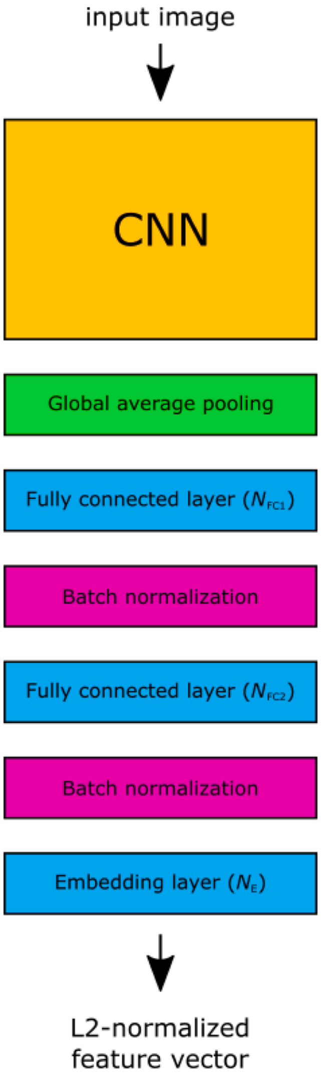

Fig. 5. Schematic of the structure of the CNN-based embedding model used in this work.

process by 0.1 and restarted the triplet-forming process for that epoch. This allowed also *easy* triplets, whose contribution to the loss was zero with the embedding model at the beginning of the epoch, to be introduced into the set of training samples. The procedure of increasing ε' and restarting the triplet-forming process was repeated until each pair (a, p) has been extended into a triplet, and then we trained the network for one epoch. The parameter ε' was reset to the value ε at the beginning of the next epoch.

2.3.3. *Deep learning approach for logs at fixed rotation*

For developing the deep learning identification method for logs at fixed rotation, we adopted the following training procedure. We performed the training in mini-batches, using a total of 1184 triplets for each epoch. At the beginning of each epoch, we agumented the training data via random in-plane rotations of approximately -18° to +18°, slight random zooming and shearing, and random changes in contrast. We did not augment the dev set. At the end of each epoch, we computed the identification accuracy separately over the (augmented) training set and the dev set, as described above. We used the Adam optimizer [31] for stochastic gradient descent.

Using the ResNet50V2 [32] CNN pre-trained on ImageNet [33], we first performed an initial scanning of hyperparameters (learning rate of the Adam optimizer = 1e-2, 1e-3, …, 1e-6; N_E = 64, 512; triplet loss margin = 0.1, 0.25, 0.5; batch size = 64, 128, 256) using N_{FC1} = 1024, N_{FC2} = 512, and 100 epochs. The goal of the hyperparameter scan was to maximize identification accuracy on the dev set. For each set of parameter values, we considered three variations: freezing all layers of the CNN, freezing all layers of the CNN except the batch normalization layers, and training all the layers of the CNN. All layers following the CNN in the embedding model were always trained. We obtained best results with the two latter variations. We then tested, for a few of the top hyperparameter sets for each of these two approaches, which training procedure was best: 1) training the entire network at once 2) first training only the embedding and batch normalization layers, and then continuing with fine-tuning the entire network. We found little difference in dev set accuracy between these two approaches, and therefore, for simplicity, we adopted the direct approach 1). In these initial runs, we used an image input size of 224 by 224 pixels. We initialized the CNNs with ImageNet pre-trained weights in all runs throughout the work.

Using the best ResNet50V2 training parameters from these initial hyperparameter tests (learning rate = 1e-4, triplet loss margin = 0.5, N_E = 512, batch size = 64), we then tested the effect of adding a dropout layer after each batch normalization layer and increasing the input size of the image. In these tests, we trained the model for 100 epochs. We found dropout to decrease model performance, but increasing the image size from 224 by 224 pixels to 512 by 512 pixels resulted in a significant increase in identification accuracy. Therefore, we adopted the image size of 512 by 512 pixels.

Then, training for 50 epochs using the best hyperparameter values from the ResNet50V2 tests, but dropping batch size to 32 due to memory constraints, we performed a comparison between different CNNs for finding the best feature extractor for the identification task (Fig. 6). The networks we considered were Xception [34], VGG-19 [35], ResNet152V2 [32], InceptionV3 [36], InceptionResNetV2 [37], MobileNetV2 [38], DenseNet201 [39], and EfficientNetB4 [40]. We found the best CNN to be InceptionResNetV2.

After this, we performed a more fine-grained hyperparameter search using InceptionResNetV2 as the CNN component of the model (learning rate = 1e-3, 1e-4, 1e-5; $N_{\mathrm{E}} = 256, 512, 1024$; triplet loss margin = 0.25, 0.50, 0.75; $N_{\mathrm{FC1}} = 1024, 2048$; $N_{\mathrm{FC2}} = 1024, 2048$; batch size = 16, 32). We found the best hyperparameter values to be the following: learning rate = 1e-4, triplet loss margin = 0.75, $N_{\mathrm{FC1}} = 1024$, $N_{\mathrm{FC2}} = 2048$, $N_{\mathrm{E}} = 1024$, batch size = 32. Next, we trained the final model on the training data using this set of hyperparameters for 120 epochs, which we deemed a good choice based on monitoring the dev set accuracy as a function of epoch (Fig. 7). Then, we evaluated the final model on the held-out test set.

2.3.4. Deep learning approach for logs at random rotation

For developing the deep learning identification method for logs at random rotations, we adopted the same CNN feature extractor and final hyperparameter values as for the case of fixed rotation. We trained the final model on the training set, using the same triplet-loss scheme as in the case of logs at fixed rotation, for 30 epochs, which we deemed a good choice based on monitoring the dev set accuracy as a function of epoch (Fig. 7). We executed image augmentation during training similarly to the case of fixed rotation, with the exception that random rotations were now performed over the full range of 0 to 360°. In addition, we permanently and randomly rotated the dev and test set images in the range of 0 to 360° prior to model development. After training the final model, we evaluated the model on the held-out test set.

The source code for developing and assessing the described CNN-based identification methods is publicly available at https://github.com/ekholmst/ultra_cnn.

2.3.5. Classical benchmark: Log identification through comparing local binary pattern histograms

As a classical benchmark for our deep learning approach, we implemented log identification using the local binary pattern (LBP) operator [23,24,41]. Out of the various published forms of LBP, we considered two: 1) the original, grayscale-invariant operator $LBP_{P,\ R}$ as presented in Ojala [24], and 2) the rotation and grayscale-invariant operator $LBP^{riu2}_{P,R}$ leveraging “uniform” LBP patterns, as presented in Ojala [24]. The LBP method has been very successful in face recognition tasks [23] as well as in other texture classification problems [41].

At its root, an LBP operator quantifies the local texture around a given image pixel using a scalar, numerical descriptor. The $LBP^{riu2}_{P,R}$ operator, which focuses on “uniform” partterns, can be thought of as a low-level feature extractor for edges, spots, flat areas, and corners [24]. When determining whether two images depict the same texture, one applies the LBP operator to each pixel in each image, computes the histogram over the occurrencies of the resulting values, and then compares the histograms of the two images using a suitable similarity measure. By dividing the image into a grid of cells, computing the LBP histogram for each cell, and then concatenating the cell-specific histograms into a single descriptor for the full image, a higher degree of spatial information can be included in the feature vector [23].

Our LBP approach was the following. First, we read in an image, transformed it into grayscale, and converted the image to size 512 by 512 pixels. Then, using the scikit-image [42] implementation of LBP, we computed the LBP result for the entire image. Next, we split the image into a grid of equally-sized cells. For each cell, we computed the histogram of LBP values, and normalized the histogram so that the sum of occurrencies in the cell was equal to one. Finally, we concatenated these cell-specific, normalized histograms to produce the final feature vector for the image. In this work, for the similarity measure for comparing two feature vectors, we adopted the (unweighted) Chi-square distance [23]

$$\chi^2(x,\ y) = \sum_{i=1}^{n} \frac{(x_i - y_i)^2}{(x_i + y_i)} \tag{3}$$

where x and y are the two feature vectors, each having length n. Our full LBP approach is illustrated in Fig. 8.

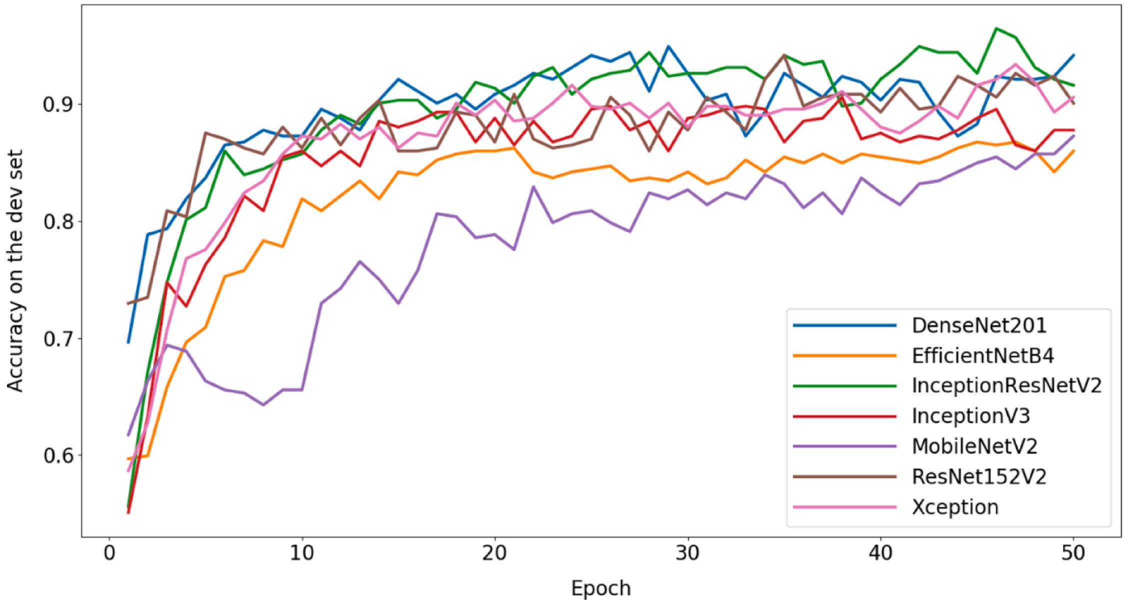

Fig. 6. Rank-1 identification accuracy on the dev set for the different CNNs in the training runs used to find the best-performing CNN feature extractor. Accuracy obtained using VGG-19, at approximately 0.2, was much lower than for the other CNNs, and was omitted from this plot for clarity.

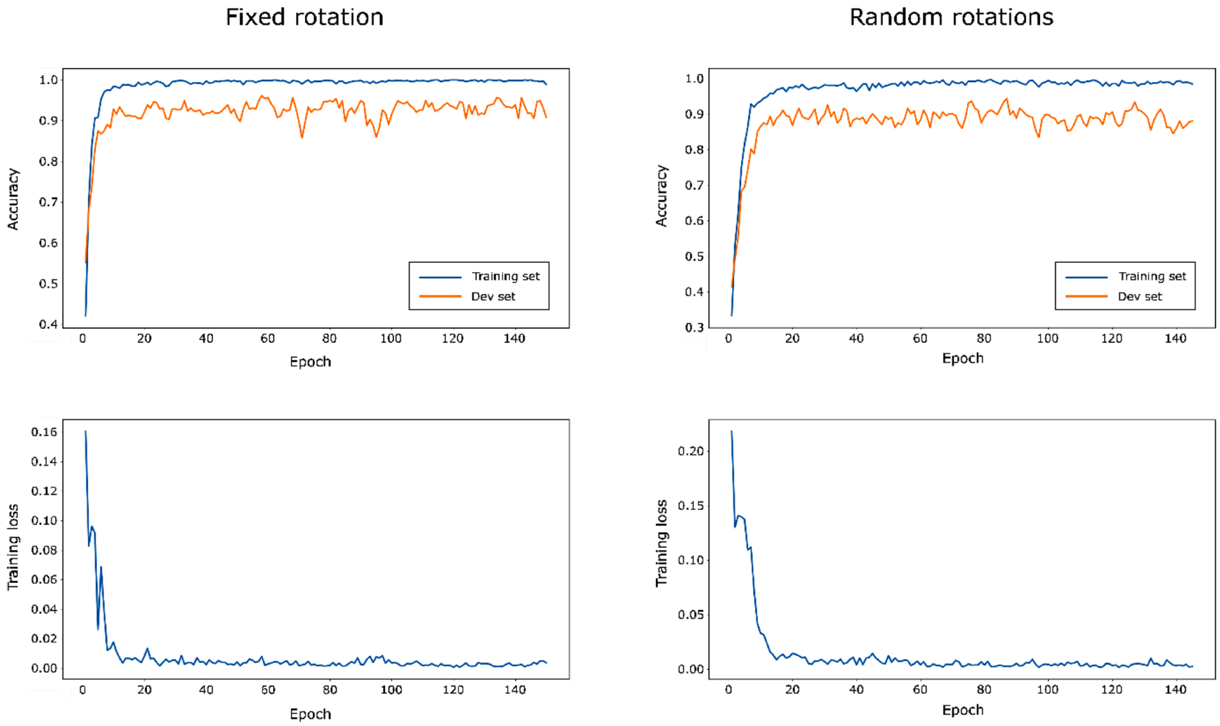

Fig. 7. Rank-1 identification accuracy and loss during training of the final model, based on InceptionResNetV2, for the case of logs at fixed rotation (left) and logs at random rotations (right). The loss at each epoch is the mean loss over minibatch-specific mean losses.

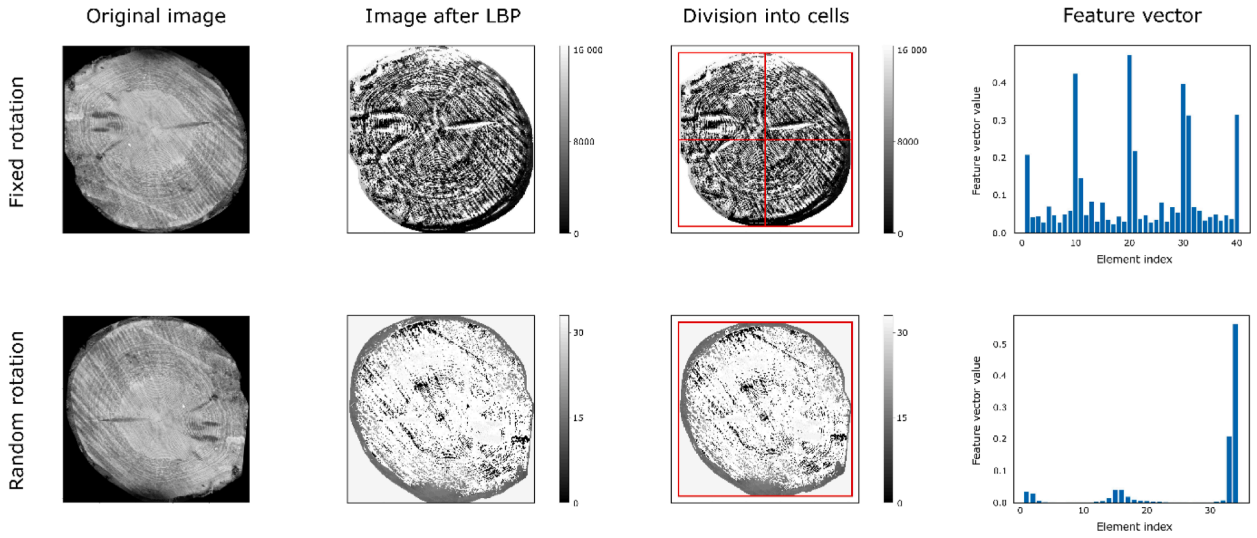

Fig. 8. Illustration of the LBP approach adopted in our work, for the case of logs at fixed rotation (upper row) and logs at random rotations (lower row), using the optimized hyperparameter values for the two cases. The image to be processed was first read in and converted into grayscale. Then, the LBP operator was applied to the entire image. Next, the LBP-filtered image was divided into equally-sized cells (2 by 2 in the upper row, 1 by 1 in the lower row). Then, the histogram of LBP pixel values was computed and normalized separately for each cell. Here the number of bins for the case of fixed log rotation was set to 10 for clarity. Finally, the histograms for the separate cells were concatenated to produce the feature vector for the image.

For LBP method 1), we tuned the following hyperparameters on the dev set: radius of the circularly symmetric local neighborhood (R), the number of points along this circle (P), the number of grid cells in both dimensions (N_{cells}), and the number of bins in the histogram for each cell (N_{bins}). The histogram for each cell consisted of equally-sized bins for values running from 0 to 2^P-1, which corresponded to the range of possible unique values produced by this operator. To tune these hyperparameters, we performed a series of grid searches. The first grid search consisted of the range $N_{\text{cells}} = 1, 3, 5, 10$; $P = 8, 16, 24, 28, 30$; $R = 1, 3, 5, 10, 20, 30$; $N_{\text{bins}} = $ 1e2, 1e3, 1e4, 1e5. When the single best set of parameter values for a single grid search was found, we performed a more detailed grid search in the vicinity of this point in parameter space. We repeated this process until the identification accuracy on the dev set stabilized. In addition to these runs, we also explored using $P = 32$ with this LBP method, which resulted in remarkably good results for some parameter sets. However, using $P = 32$ always led to numerical overflow of the LBP implementation, and many cases suffered from instability. Therefore, we constrained P to a maximum value of 30 for this LBP method.

For method 2), N_{bins} was always set to P+2 to correspond to the number of possible unique values given by this variant of the LBP operator. The histogram for each cell here consisted of equally-sized bins for values running from 0 to P+1. The initial range of values in the hyperparameter search for this method was $N_{\text{cells}} = 1, 3, 5, 10$; $P = 8, 16, 32, 64$; $R = 1, 3, 5, 10, 20, 30$.

For logs at fixed rotation, we obtained the best results on the dev set when using the original LBP method 1) with a grid size of 2 by 2 cells, $P = 14$, $R = 15$, and $N_{\text{bins}} = 1e4$. Using this LBP setup, we finally applied the method on the held-out test set of logs at fixed rotation.

For logs at random rotation, i.e., using the dev set where logs had been permanently and randomly rotated, we obtained the best results for identification when using the "uniform" LBP method 2) with a grid size of 1 by 1 cells, $P = 32$, and $R = 80$. Finally, we evaluated this method on the held-out test set of randomly rotated logs.

The source code for developing and assessing the described LBP-based methods is publicly available at https://github.com/ekholmst/ultra_cnn.

3. Results

Identification accuracies obtained using the final deep learning models and the LBP approaches on the dev set and the held-out test set are given in Table 1. For both considered cases of log rotation, i.e., fixed or random, the CNN greatly outperforms the LBP benchmark. For random rotations, the LBP approach results in remarkably poor performance. The CNN methods exhibit slight overfitting for both considered cases of log rotation, seen as a drop of 3-6% in accuracy when moving from the dev set to the test set. The CNN identification result is slightly better in the case of fixed rotation as compared to random rotations.

Case examples of identification results for logs at fixed and random rotation on the test set are given in Figs. 9 and 10, respectively. The distance between the embedding vector of the probe image and those of the registered templates is plotted in these Figures to further illustrate the identification process. The successful identifications in Figs. 9 and 10 include examples where lighting, imaging angle, and perceived coloring of the log, likely due to aging, varies significantly between the probe image and the correct template. In the examples of failed identifications, the erroneously matched logs display mutually similar features. For these cases, the plots of embedding distances reveal that although the rank-1 identification failed, the correct template log was within the first few closest embedding vectors to the probe image.

Among the 396 identification queries on the test set (four probe images for each of the 99 unique logs), 23 were ones that failed for both fixed and random rotations. Within these, there were three logs for which the identification failed for all four probe images of the log (Fig. 11). One of the logs was a butt log with a prominent residue of what was most likely stump treatment liquid, whereas the other two were relatively plainly featured logs from higher up the stem. There were 41 queries where the identification at fixed rotation was successful but failed when the images were randomly rotated. Interestingly, for a total of 14 queries, the identification at random rotation was successful but failed for the same images at fixed rotation.

Table 1
Rank-1 identification accuracy (%) for the final CNN and LBP approaches when attempting to match angled, aged shots with fresh headshots of logs (see Fig. 4). The error is the standard error of the accuracy. There were 99 logs in the test set, with four probe images for each, totaling 396 identification queries. Evaluation on the 98 dev set logs comprised 392 identification queries. Choosing the best match for a given probe image uniformly randomly corresponds to an identification accuracy of approximately 0.01.

	Logs at fixed rotation		Logs at random rotations	
	CNN	LBP	CNN	LBP
dev set	93.4 ± 1.3	72 ± 2	89.8 ± 1.5	15 ± 2
test set	90.7 ± 1.5	67 ± 2	84 ± 2	12 ± 2

In the current implementation, computing the CNN embedding for a single 512 x 512 RGB image takes approximately 1 s on a single CPU core (Intel Xeon Gold 6230 @ 2.1 GHz). This is likely too slow for real forest operations, but the time could be significantly reduced by performing the computation on an embedded GPU. The time required for performing the identification process for a single probe image on the database of 99 fresh headshot images of the test set takes approximately 0.25 ms. Finding the closest-matching embedding via Euclidean distance is probably not a performance bottleneck for the identification process, even with log pools of realistic size.

To assess how the identification accuracy of our CNN approach behaves as a function of log pool size, we computed the accuracy of identifying the test set logs using template databases of different sizes. To extend the size of the template database beyond the test set log fresh headshots, we successively added fresh headshot images from the training and dev sets to the pool of templates. As our identification models had been subjected to the training and dev set images during training, this procedure does not give an unbiased estimate of the accuracy when using logs beyond those of the test set in the database. The results imply, however, that the identification accuracy falls as a function of log pool size, as expected (Fig. 12). The decline is sharper for logs at random rotations than in the case of fixed rotation, where the decline appears to be weak.

4. Discussion

The estimate for the rank-1 identification accuracy of our deep learning approach on a pool of 300 logs, (89 ± 2)% for logs at fixed rotation (Fig. 12), is slightly lower than the accuracy of (93.6 ± 1.3)% reported by Schraml et al. [7] on a pool of 279 logs. In Schraml et al. [7], rotational pre-alignment was performed to set all the images of a given log to approximately the same orientation. For composing the feature vector for a log end image, the authors of Schraml et al. [7] employed a classical image analysis method based on Gabor filtering. However, the dataset in their work appears to contain little variation between the different images of a given log, the variance mainly coming from either using or not using flash when photographing a log. In contrast, in our dataset, the different images of a log are subject to strong changes in perspective, often in lighting as well, with the additional effect of five days of aging in between the probe and template images of each log. Also, the images in our approach were automatically segmented using a CNN, whereas in Schraml et al. [7], the segmentation was done manually. The slightly lower accuracy of our CNN as compared to the classical approach of Schraml et al. is most likely due to the more challenging dataset in our work and differences in the image preparation procedure. To gauge the importance of the perspective changes for identification accuracy, we computed identification accuracy for aged headshot probe images matched against fresh headshot template images using the test set. The result was (94 ± 2)% and (91 ± 3)% for fixed and random orientations, respectively, somewhat higher than for the case of using angled shots for the probe images ((90.7 ± 1.5)% and (84 ± 2)%, respectively, Table 1).

Wimmer et al. [20] presented a two-staged method for log identification, which was similar to our approach: first segmenting the log end from a given image using a CNN, and then performing identification using CNN-based feature embeddings. Their dataset consisted of a total of 379 unique logs, of which 279 were the same as in Schraml et al. [7] and 100 were new. For the 100 new logs, the images had varying perspectives, different rotations, raw and sanded surfaces, different cameras employed, and thin discs cut off from each end of the log, in addition to images of the intact log ends. The authors concluded that their CNN approach performed much better than the classical approach of Schraml et al. [7]. This is similar to our findings of CNN performance compared to the new LBP benchmark. In addition, the authors

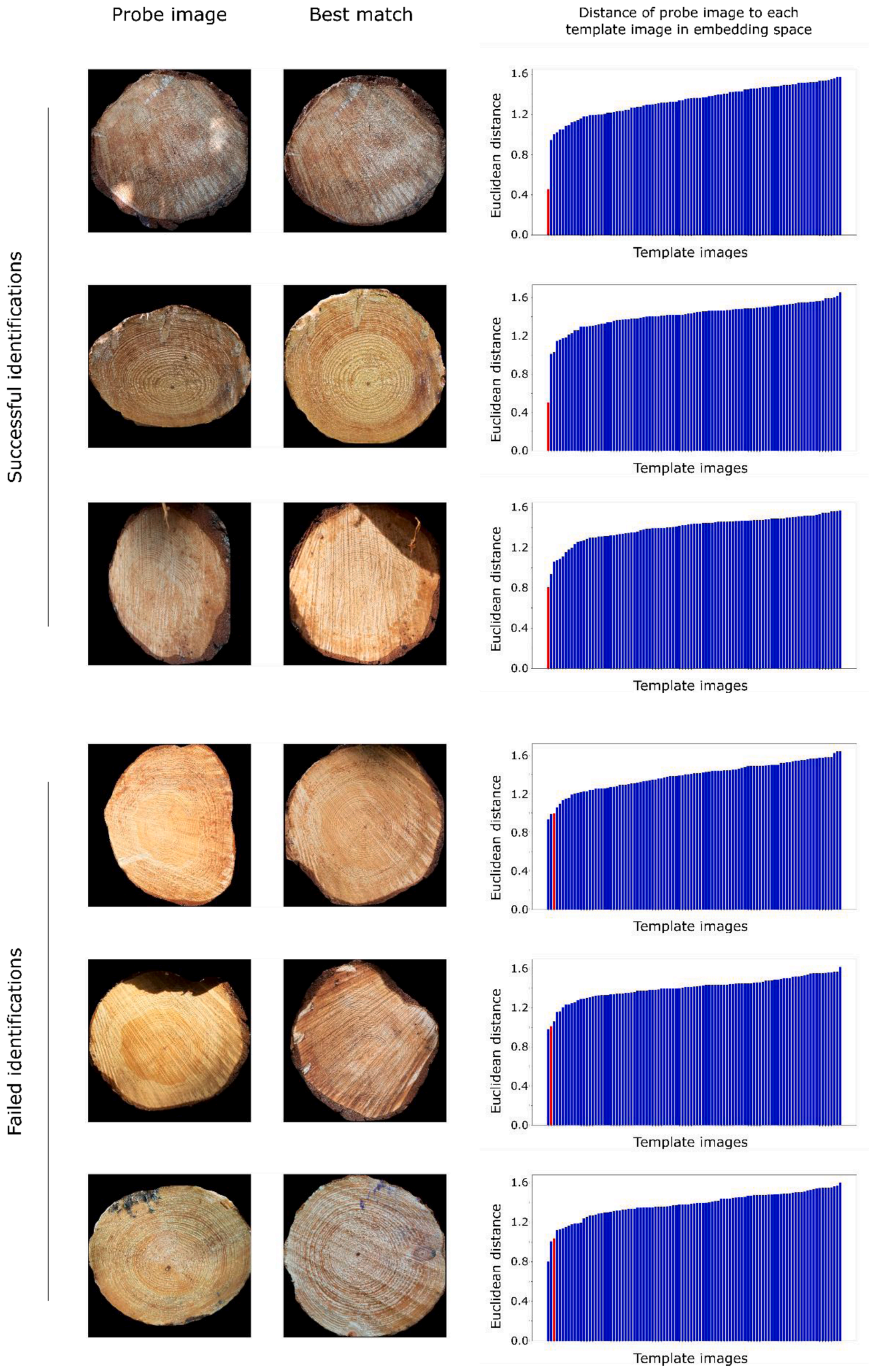

Fig. 9. Examples of successful and failed identifications when applying the final CNN model to the test set of logs at fixed rotation. The probe image in the first column depicts an aged, angled shot of a log, whose fresh headshot image was to be found in the database of registered templates. The best match in the database found by the identification algorithm is shown in the second column. The third column shows the distance of the probe image embedding to the embedding of each template, the red bar signifying the headshot of the probe image log in the database.

concluded that rotational pre-alignment was not necessary for performing log identification. Our results, however, imply that analyzing the log images in approximately the same rotational orientation improves the identification result by a modest but significant amount, and involves much slower deterioration of accuracy with growing template database size (Table 1, Fig. 12).

As regards the LBP benchmark itself, the default form of the texture operator, applied on a square grid of 2 by 2 cells over the full end face image, produces fairly good results for identification of logs at fixed rotation. The operator seems to be able to capture uniquely identifying textures in different regions of a given log end face. Indeed, the spatial information available from the partitioning of the log end into grid cells appears to be necessary for attaining good results with LBP on this problem. This observation is supported by the optimum of 2 by 2 cells

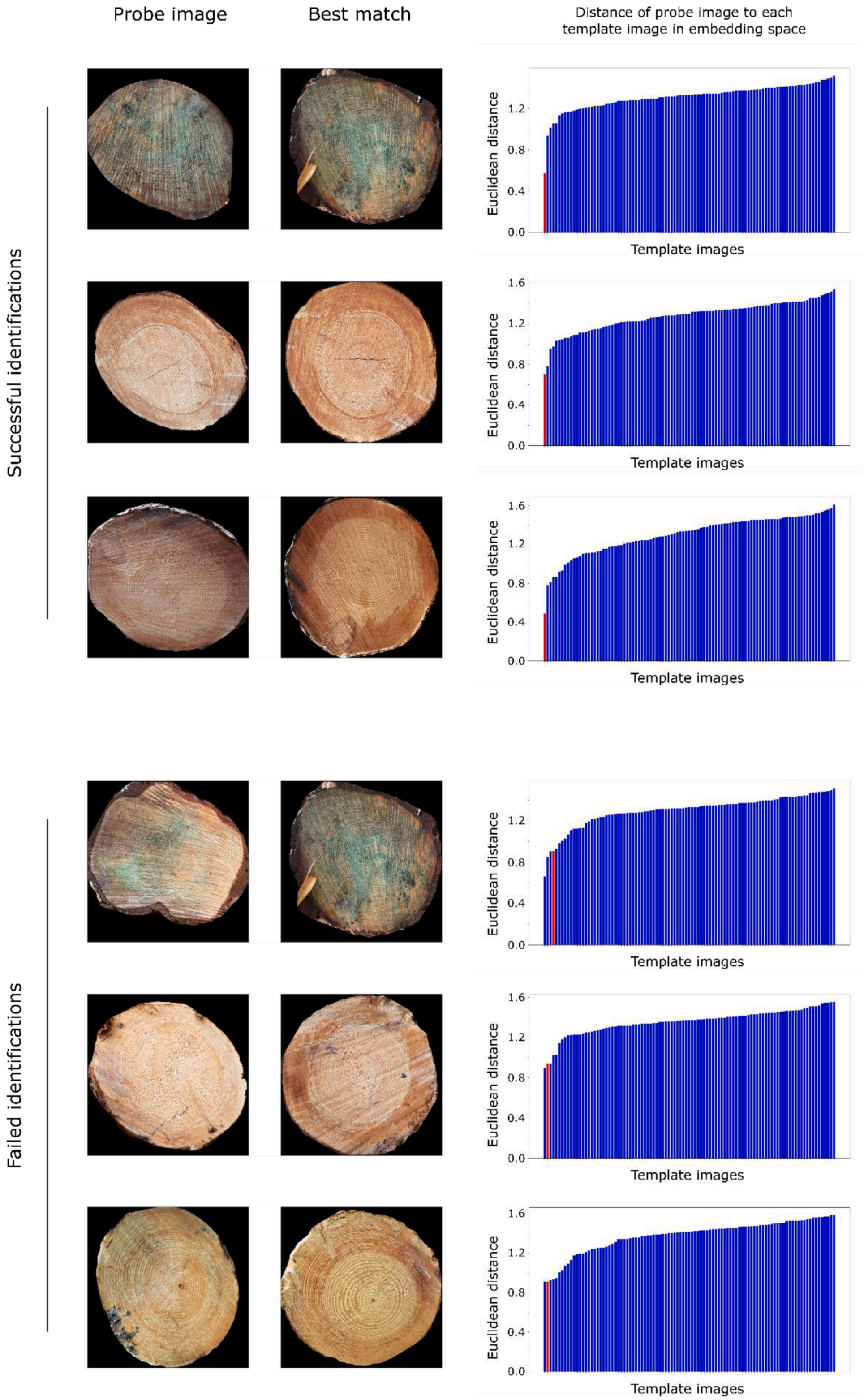

Fig. 10. Examples of successful and failed identifications when applying the final CNN model to the test set of logs at random rotations. The probe image in the first column depicts an aged, angled shot of a log, whose fresh headshot image was to be found in the database of registered templates. The best match in the database found by the identification algorithm is shown in the second column. The third column shows the distance of the probe image embedding to the embedding of each template, the red bar signifying the headshot of the probe image log in the database.

instead of a single cell for the entire image, and the fact that we obtained the lowest identification accuracies using a single cell for the entire image. For the case of randomly rotated logs, partitioning the image into cells does not make sense, as a given cell will typically encapsulate completely different parts of the log face as the log face is rotated between the template and probe image. Indeed, here we found the best results by using a single cell for the entire image. As the spatial information regarding the textures in different regions of the log face is not available in this case, the identification accuracy is very low.

Both the identification accuracy and the query time can be expected to deteriorate as the size of the database of registered log templates grows. In real-world operations, the full database of templates might typically be very large. However, the database could likely be trimmed down to a relevant subset of registered logs, depending on the specific

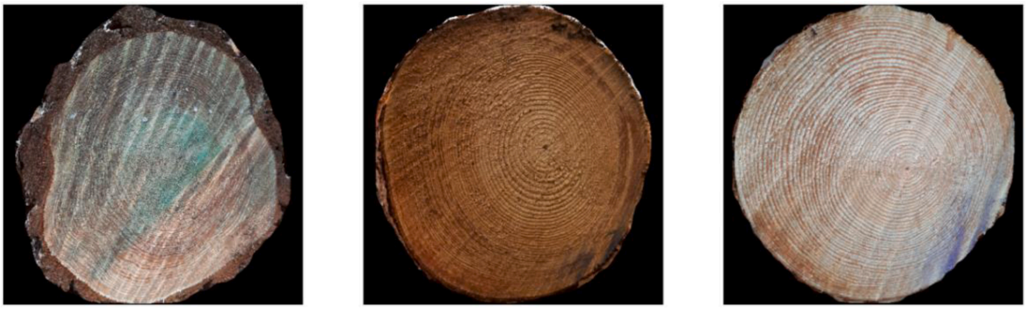

Fig. 11. Example probe images of the three logs in the test set for which identification failed at both fixed and random rotations.

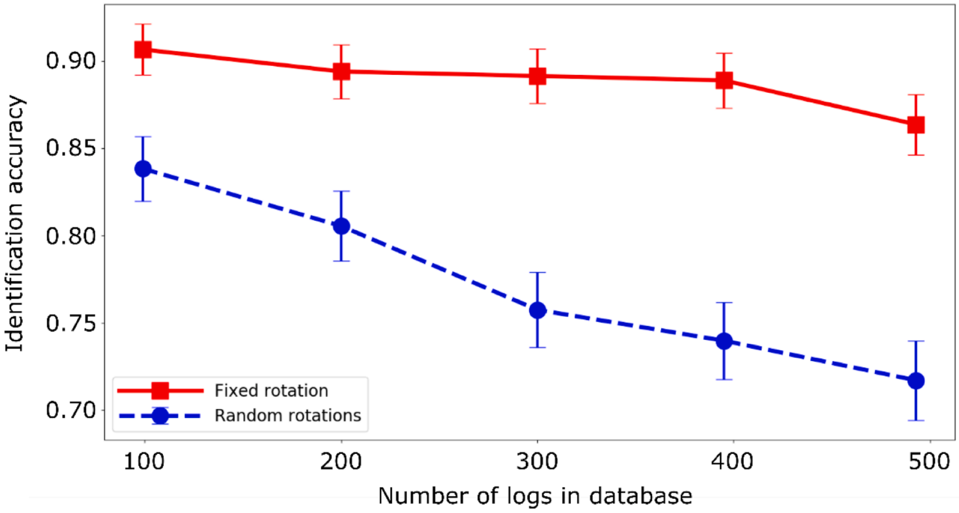

Fig. 12. Rank-1 identification accuracy when matching the 396 aged, angled shots of the 99 test set logs against fresh headshot pools of different sizes. Pools of more than 99 logs were obtained by adding logs from the training and dev sets to the database. The errorbar is the standard error of the accuracy.

identification use case. For example, when identifying individual logs arriving at the sawmill, if the harvest site from which the timber truck picked up the batch of logs was known, only the registered logs from that harvest site would need to be considered as candidate templates. This would bring the number of candidate logs down to perhaps a few thousand. In the case of identifying stored logs at the sawmill yard, the relevant pool of registered logs could be perhaps constrained to, e.g., logs from the 25-30 most recent harvest sites. This could bring the number of candidate logs down to the order of 10^5.

In more general identification cases, these approaches to constraining the set of candidate templates may not always be possible. However, in any use case, the database could possibly be "pre-filtered" to consider only template logs with similar external geometrical features, e.g., log face area and height to width ratio, as the log in the probe image [43]. Yet another approach might be to cluster logs via unsupervised machine learning using their deep feature vectors [43]. Through ideas such as the ones presented here for realizing the informed sampling of the database of templates, we believe the CNN approach would be feasible in real forest operations.

To further increase identification accuracy, structural attributes of the logs, such as the diameter profile, measured first by the harvester and measured again when the logs arrive at the sawmill, could be used to augment the image-based identification as additional features to use in the matching. As another example, the branch structure of standing trees, as deduced from terrestrial laser scanning data [44–46] possibly available through harvester-mounted devices in the future, and the knot structure of logs, as determined via X-ray tomography upon arriving at the sawmill [5], could be used to construct yet more features for strengthening the identification.

Even if the rank-1 identification accuracy in real-world applications was much less than 100%, a scenario for using such an imperfect rank-1 identification method for tackling the illegal sourcing of wood could be the following. First a log, along with its harvest site of origin, is queried from the database. If the returned location points to an unauthorized harvest site, a warning is given, and the log is "flagged" indicating that it is suspected to be illegally sourced. Similarly, if no matching log within a pre-determined threshold of the chosen similarity measure (Euclidean distance, etc.) is found in the database, a warning is given, and the log is "flagged" as possibly originating from an unknown source.

Furthermore, when constructing datasets of wood appearance labeled by wood quality, i.e., appearance of freshly harvested log ends and the detailed quality information for the corresponding logs as determined at the sawmill, for the purpose of creating predictive models for wood characteristics, it may not be necessary to correctly match the appearance-quality information for the exact same log. Instead, labeling a given log image with the quality information of a log of *similar* appearance, even if not the exact same log, may well be enough to enable the creation of such wood quality models.

Determining the harvest site of origin for a log might also be accomplished as follows. First, for each harvest site, one computes an effective, average feature vector for the set of logs originating at that site. Then, for finding the site of origin for a given individual log, one determines which such mean feature descriptor is closest to that of the log. Such a single-to-batch matching approach could perhaps be used alongside individual log identification to label the tracing result reliable when both methods point to the same harvest site of origin for the log.

In light of our results, the identification of individual logs through RGB images of the log end appears a promising approach to tracking and tracing logs over the wood procurement chain. However, to further boost identification accuracy, it may become helpful to leverage information from beyond the visible spectrum. Recent results by Schraml et al. [47] suggest that the near-infrared reflectance spectral bands of log ends contain information with potential for strongly discriminating individual logs from one another using a few selected wavelengths. Instead of employing a full hyperspectral camera for log identification, such results could be used to guide the design and creation of simpler and more affordable multispectral imaging and analysis systems targeting only the most critical wavelengths for log identification.

5. Conclusions

In this work, using a large and diverse dataset of log end images, we developed and assessed CNN-based methods for tree log identification. Our results show that the deep learning approach appears feasible, even when aging of logs, different lighting, and varying imaging angles are considered. Deep CNNs greatly outperform a powerful, classical texture-based method on the task, implemented here as a benchmark. Allowing for random rotations of the log end makes the identification more challenging. Due to the decrease in identification accuracy with increasing log pool size, pre-filtering or clustering of the database of registered templates will likely be needed to make the deep learning approach applicable to real forest operations. We publish the full dataset of 4930 log end images created for this work.

CRediT authorship contribution statement

Eero Holmström: Methodology, Software, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization, Funding acquisition. **Antti Raatevaara:** Methodology, Software, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Funding acquisition. **Jonne Pohjankukka:** Methodology, Writing – review & editing. **Heikki Korpunen:** Conceptualization, Investigation, Writing – review & editing, Funding acquisition. **Jori Uusitalo:** Conceptualization, Investigation, Writing – review & editing.

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.

Data availability

We have published the data and the code used for this work.

Acknowledgments

The authors wish to thank UPM Oyj and Metsäkonepalvelu Oy for their generous assistance in organizing the data acquisition for this work and for fruitful discussions. In addition, the authors gratefully acknowledge CSC – IT Center for Science, Finland, for computational resources. EH wishes to thank Lauri Sikanen, Mikko Peltoniemi, Niko Kulha, and Tuomas Yrttimaa for discussions on the challenges and possibilities of log identification. EH also wishes to thank Tuula Piri and Elina Roininen for further Scots pine samples. This work was funded by the ULTRA project (no. 41007-00216900) at Luke, and the Academy of Finland Flagship "Forest-Human-Machine Interplay – Building Resilience, Redefining Value Networks and Enabling Meaningful Experiences" (funding decision 337655).

References

[1] D. Sandberg, et al., The role of the wood mechanical industry in the Swedish forest industry cluster, Scand. J. Res. 29 (4) (2014) 352–359.
[2] A. Ostovar, B. Talbot, S. Puliti, R. Astrup, O. Ringdahl, Detection and classification of root and butt-rot (RBR) in stumps of Norway Spruce using RGB images and machine learning, Sensors 19 (7) (2019) 1579.
[3] A. Raatevaara, H. Korpunen, H. Mäkinen, J. Uusitalo, Log end face image and stem tapering indicate maximum bow height on Norway spruce bottom logs, Eur. J. Res. 139 (6) (2020) 1079–1090.
[4] D. Kleinschmit, S. Mansourian, C. Wildburger, A. Purret, Illegal Logging and Related Timber Trade-Dimensions, Drivers, Impacts and Responses. A Global Scientific Rapid Response Assessment Report, 35, IUFRO (International Union of Forestry Research Organizations) Secretariat, 2016.
[5] H. Mäkinen, H. Korpunen, A. Raatevaara, J. Heikkinen, J. Alatalo, J. Uusitalo, Predicting knottiness of Scots pine stems for quality bucking, Eur. J. Wood Wood Prod. 78 (1) (2020) 143–150.
[6] K. Marjanen, P. Ojala, H. Ihalainen, Measurement of annual ring width of log ends in forest machinery, in: Image Processing: Algorithms and Systems VI, 6812, International Society for Optics and Photonics, 2008, p. 68120D.
[7] R. Schraml, H. Hofbauer, A. Petutschnigg, A. Uhl, On rotational pre-alignment for tree log identification using methods inspired by fingerprint and iris recognition, Mach. Vis. Appl. 27 (8) (2016) 1289–1298.
[8] G. Rune, M. Warensjö, Basal sweep and compression wood in young scots pine trees, Scand. J. Res. 17 (6) (2002) 529–537.
[9] G.I. Vestøl, C. Fischer, O. Høibø, Simulation of structural timber properties based on geographical data and stand-level forest inventory data, Scand. J. For. Res. 35 (5–6) (2020) 286–295.
[10] L. Wilhelmsson, J. Arlinger, K. Spångberg, S.O. Lundqvist, T. Grahn, Ö. Hedenberg, L. Olsson, Models for predicting wood properties in stems of picea abies and pinus sylvestris in Sweden, Scand. J. Res. 17 (4) (2002) 330–350.
[11] E. Duchateau, F. Longuetaud, F. Mothe, C. Ung, D. Auty, A. Achim, Modelling knot morphology as a function of external tree and branch attributes, Can. J. Res. 43 (3) (2013) 266–277.
[12] H. Mäkinen, F. Colin, Predicting branch angle and branch diameter of Scots pine from usual tree measurements and stand structural information, Can. J. Res. 28 (11) (1998) 1686–1696.
[13] J. Pyörälä, et al., Assessing log geometry and wood quality in standing timber using terrestrial laser-scanning point clouds, Int. J. Res. 92 (2) (2019) 177–187.
[14] J. Uusitalo, Pre-harvest measurement of pine stands for sawing production planning, Ph.D. Thesis, The Society of Forestry in Finland - Finnish Forest Research Institute, 1997.
[15] D.P. Dykstra, G. Kuru, R. Taylor, R. Nussbaum, W.B. Magrath. Technologies for wood tracking: verifying and monitoring the chain of custody and legal compliance in the timber industry, World Bank, Washington, DC, 2002.
[16] F. Müller, D. Jaeger, M. Hanewinkel, Digitization in wood supply–a review on how Industry 4.0 will change the forest value chain, Comput. Electron. Agric. 162 (2019) 206–218.
[17] B. Bhanu, A. Kumar, Deep Learning for Biometrics, Springer, Cham, 2017.
[18] H.E. Khiyari, H. Wechsler, Age invariant face recognition using convolutional neural networks and set distances, J. Inf. Secur. 8 (03) (2017) 174.
[19] O.M. Parkhi, A. Vedaldi, A. Zisserman, Deep face recognition (2015).
[20] G. Wimmer, R. Schraml, H. Hofbauer, A. Petutschnigg, A. Uhl, Two-stage cnn-based wood log recognition, in: Proceedings of the International Conference on Computational Science and Its Applications, Springer, Cham, 2021, pp. 115–125.
[21] M. Robert, P. Dallaire, P. Giguère, Tree bark re-identification using a deep-learning feature descriptor, in: Proceedings of the 2020 17th Conference on Computer and Robot Vision (CRV), IEEE, 2020, pp. 25–32.
[22] M. Vihlman, J. Kulovesi, A. Visala, Tree log identity matching using convolutional correlation networks. 2019 Digital Image Computing: Techniques and Applications (DICTA), IEEE, 2019, pp. 1–8.
[23] T. Ahonen, A. Hadid, M. Pietikainen, Face description with local binary patterns: application to face recognition, IEEE Trans. Pattern Anal. Mach. Intell. 28 (12) (2006) 2037–2041.
[24] T. Ojala, M. Pietikainen, T. Maenpaa, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Trans. Pattern Anal. Mach. Intell. 24 (7) (2002) 971–987.
[25] Y. Wu, A. Kirillov, F. Massa, W.Y. Lo, R. Girshick, Detectron2, 2019. https://github.com/facebookresearch/detectron2.
[26] K. He, X. Zhang, S. Ren, J. Sun, Identity mappings in deep residual networks, in: Proceedings of the European Conference on Computer Vision, Springer, Cham, 2016, pp. 630–645.
[27] K. Wada, Labelme: Image polygonal annotation with python, 2018. https://github.com/wkentaro/labelme.
[28] F. Chollet, et al., Keras, 2015. https://keras.io).
[29] M. Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
[30] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: A unified embedding for face recognition and clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 815–823.
[31] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).
[32] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[33] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, et al., Imagenet large scale visual recognition challenge, Int. J. Comput. Vis. 115 (3) (2015) 211–252.

[34] F. Chollet, Xception: Deep learning with depthwise separable convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1251-1258..

[35] K. Simonyan, and A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).

[36] C. Szegedy, et al., Rethinking the inception architecture for computer vision, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[37] C. Szegedy, et al., Inception-v4, inception-resnet and the impact of residual connections on learning, in: Proceedings of the Thirty-first AAAI Conference on Artificial Intelligence, 2017.

[38] M. Sandler, et al., Mobilenetv2: Inverted residuals and linear bottlenecks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018.

[39] G. Huang, et al., Densely connected convolutional networks, Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4700-4708.

[40] M. Tan, Q. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, in: Proceedings of the International Conference on Machine Learning. PMLR, 2019.

[41] T. Ojala, M. Pietikäinen, D. Harwood, A comparative study of texture measures with classification based on featured distributions, Pattern Recognit. 29 (1) (1996) 51–59.

[42] S. Van der Walt, et al., Scikit-image: image processing in Python, PeerJ 2 (2014) e453.

[43] N. Rajabli, Improving biometric log detection with partitioning and filtering of the search space, M.Sc. Thesis, Linnaeus University, 2021.

[44] J. Hackenberg, et al., Highly accurate tree models derived from terrestrial laser scan data: a method description, Forests 5 (5) (2014) 1069–1105.

[45] J. Pyörälä, et al., Comparison of terrestrial laser scanning and X-ray scanning in measuring Scots pine (Pinus sylvestris L.) branch structure, Scand. J. Res. 33 (3) (2018) 291–298.

[46] J. Pyörälä, et al., Quantitative assessment of Scots pine (Pinus sylvestris L.) whorl structure in a forest environment using terrestrial laser scanning, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 11 (10) (2018) 3598–3607.

[47] R. Schraml, K. Entacher, A. Petutschnigg, T. Young, A. Uhl, Matching score models for hyperspectral range analysis to improve wood log traceability by fingerprint methods, Mathematics 8 (7) (2020) 1071.