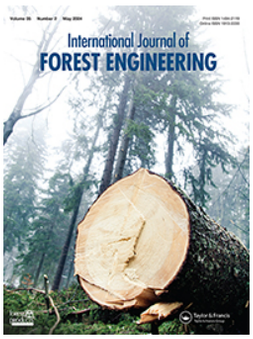

International Journal of Forest Engineering

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/tife20

Automatic detection of root rot and resin in felled Scots pine stems using convolutional neural networks

Eero Holmström, Henna Kainulainen, Antti Raatevaara, Jonne Pohjankukka, Tuula Piri, Juha Honkaniemi, Jori Uusitalo, Mikko Peltoniemi & Aleksi Lehtonen

To cite this article: Eero Holmström, Henna Kainulainen, Antti Raatevaara, Jonne Pohjankukka, Tuula Piri, Juha Honkaniemi, Jori Uusitalo, Mikko Peltoniemi & Aleksi Lehtonen (2024) Automatic detection of root rot and resin in felled Scots pine stems using convolutional neural networks, International Journal of Forest Engineering, 35:2, 153-165, DOI: 10.1080/14942119.2024.2327247

To link to this article: https://doi.org/10.1080/14942119.2024.2327247

Published online: 14 Mar 2024.

INTERNATIONAL JOURNAL OF FOREST ENGINEERING
2024, VOL. 35, NO. 2, 153–165
https://doi.org/10.1080/14942119.2024.2327247

Automatic detection of root rot and resin in felled Scots pine stems using convolutional neural networks

Eero Holmström[a], Henna Kainulainen[a], Antti Raatevaara[a,b], Jonne Pohjankukka[a], Tuula Piri[a], Juha Honkaniemi[a], Jori Uusitalo[a,c], Mikko Peltoniemi[a], and Aleksi Lehtonen[a]

[a]Natural Resources Institute Finland (Luke), Helsinki, Finland; [b]MiCROTEC Innovating Wood Oy, Espoo, Finland; [c]Department of Forest Sciences, University of Helsinki, Helsinki, Finland

ABSTRACT

Root rot caused by *Heterobasidion* spp. is the most serious fungal disease of conifer forests in the Northern Hemisphere. In Scots pine (*Pinus sylvestris L.*) stands infected by *H. annosum*, root rot reduces sawlog quality due to decay and resin-soaked patches. Automatically detecting the disease during harvesting operations could be used to optimize bucking as well as to efficiently collect data on root-rot incidence within forest stands and at larger geographical scales. In this study, we develop deep learning models based on convolutional neural networks to automatically detect root rot disease and the presence of resinous wood in stem end images of Scots pine. In addition, we study the effect of pre-filtering the images via a classical texture operator prior to model development. Using transfer learning on pre-trained feature extractor networks, we first construct classifiers for detecting severely rotten wood in stem end images. Second, we develop a classifier for detecting the presence of resin outside branch knots. In rot detection, using regular RGB images, our final model reaches a binary classification accuracy of (63 ± 6)% on the independent test data, where the error is the standard error. Pre-processing the images using the classical texture operator increases the final classification accuracy to (70 ± 6)%. To detect only resin using regular RGB images, we find an accuracy of (80 ± 6)%. Finally, we discuss the operational implications and requirements of implementing such computer vision algorithms in the next generation of forest harvesters.

ARTICLE HISTORY
Received 20 February 2023
Accepted 1 March 2024

KEYWORDS
Heterobasidion root rot; wood quality; deep learning; bucking optimization

Introduction

Heterobasidion spp. are widely occurring fungi in the coniferous forests of the Northern Hemisphere (Garbelotto and Gonthier 2013). Through root rot disease, *Heterobasidion* spp. cause significant damage to forests. In Scots pine (*Pinus sylvestris* L.), *H. annosum* sensu stricto (Fr.) Bref. is the leading agent of decay, causing growth losses and a reduction in timber yield, inflicting significant financial losses on forest owners and wood purchasers (Burdekin 1972; Wang et al. 2014). The disease spreads among the root systems of trees, eventually reaching the stem, where its signature symptom is resin secreted by the host in an attempt to contain the spread of the fungal mycelium and the ensuing decay. The resin-soaked patches and the decay at the base of the stem reduce sawlog quality. Eventually, the fungus kills the host tree (Greig 1995). Climate change is expected to increase forest disturbances due to pathogens (Müller et al. 2014; Seidl et al. 2017), which implies that the ecological and economic damage caused by *Heterobasidion* spp. will probably increase in the future.

In cut-to-length (CTL) harvesting, tree stems are bucked into logs immediately after felling. The prospect of imaging tree stems at the time of bucking via a camera mounted onto the harvester (Mäkinen et al. 2019; Ostovar et al. 2019) could make it possible to automatically detect and analyze root rot disease. Such analysis could provide the information necessary to accurately separate the diseased and healthy portions of the stem, either automatically or manually, preventing rotten wood ending up at the sawmill or healthy sawlog material being left on the forest floor in overly large amounts. The automatic detection of rot by harvesters, paired with location data, could also be used to create maps of root rot incidence at the stand level, as well as at larger geographical scales (Puliti et al. 2018). This would support the planning of precision management strategies for the efficient prevention or containment of root rot disease when regenerating stands, as well as the development of large-scale risk prediction models for the disease.

While resin is a smaller concern than decay in regards to wood quality, the presence of resin is considered a defect that reduces timber quality. Implementing a computer vision-based resin detector alongside other analysis tools in a harvester would probably incur only a minor computational cost and still provide useful information about wood quality. Automatic registration when considerable amounts of resin are present in a harvested sawlog could therefore enable more precise assessment of the quality and value of sawlogs being produced in CTL operations than is possible today.

Depending on the severity of root rot disease in Scots pine, the resin pattern visible at the stem end can be prominent or subtle, sometimes appearing similar to resin patches formed as a response to mechanical damage to the stem or due to decay agents other than *Heterobasidion* spp. In addition, the decay

CONTACT Eero Holmström eero.holmstrom@luke.fi Natural Resources Institute Finland (Luke), Latokartanonkaari 9, Helsinki 00790, Finland

and resin due to root rot do not always appear together. These complex visual aspects of the disease make it challenging to analyze root rot in a stem end image.

Convolutional neural networks (CNNs) have been used to tackle a wide range of image recognition tasks with great success (Goodfellow et al. 2016). The power of CNNs in analyzing images of wood has recently been demonstrated in, e.g. detecting root rot in images of Norway spruce stumps (Puliti et al. 2018; Ostovar et al. 2019; Nowell 2019), identifying individual logs based on their bark textures (Vihlman et al. 2019; Robert et al. 2020) or end face images (Holmström et al. 2023), and in counting annual tree rings in cross-sectional images of stems (Fabijańska and Danek 2018). CNNs are generally more robust than classical image analysis approaches with respect to imaging conditions and require very little manual tuning by the model developer, which makes them a powerful tool for solving computer vision problems.

In this study, we develop and assess CNN-based deep learning models for detecting root rot disease in stem end images of Scots pine. In developing these models, we consider both regular RGB images of log ends and the same images first filtered via a classical texture operator to determine whether such pre-processing could produce new features useful for detecting the disease. In addition, using regular RGB images, we develop and present a model for detecting when significant amounts of resin are visible on a stem end. Finally, we discuss the implications and the required performance of such models for real forest operations.

Materials and methods

Collecting and labeling the image data

To develop CNN models for the task of root rot detection, we collected an original data set comprising both healthy and infected Scots pine trees. In a single-species Scots pine stand in Southern Finland, known from an earlier survey to be severely infected by *H. annosum*, 30 trees were felled in the late fall of 2019 for this purpose. The stand, described in detail in Pitkänen et al. (2021), is originally an experiment for testing the breeding value of plus-trees (Ruotsalainen 2014) from Southern and Central Finland, all the 30 studied trees being siblings of each other with limited genetic variation. As the decay and resin from root rot typically do not reach higher than ~30 cm up into the stem of Scots pine (Laine 1976), each tree was cut as close to the ground as possible to reliably identify diseased stems. For each tree, the appearance of the cut surface was immediately assessed on felling. In the case of a suspected or clear case of root rot, disks of ~5 cm were cut from the stem until no signs of the disease were visible. The number of disks produced from each tree ranged from 1 to 26 with an average of 6.4, leading to a total of 192 disks from the total set of 30 stems.

After the felling operation, the disks were moved to a fridge for overnight storage. The following day, the bottom face of each disk was first brushed clear of sawdust and dirt, etc. and then photographed at room temperature. The camera was a Nikon D7200, the lens was a Sigma DC 17–50 mm 1:2.8 EX HSM, the working distance was approximately 50 cm, and the photography was performed under artificial lighting. The image resolution was 6000 pixels by 4000 pixels. Keeping the disks at room temperature prior to the photographing increased the contrast between resinous areas and the clean wood surface. A total of 192 disks were processed and photographed in this way.

For each of the 192 disk images, the severity of the observed decay was assessed on a scale of 0 to 4, with 0 meaning no rot and 4 indicating advanced decay (Pitkänen et al. 2021). Each disk was then assigned a binary label of either "rotten" or "healthy," with "rotten" indicating rot severity of degree 4 and "healthy" indicating any other level of rot. This relatively coarse approach was taken to simplify the binary classification task at hand in light of the small number of images and the intricacies of the visual symptoms of the disease.

An alternative labeling of the data was produced to create a classifier model for solely detecting resin in the images. For this latter classification task, each image was labeled either "contains extra resin" or "clean," depending on whether the image displayed significant amounts of extractives outside branch knots. In addition to resin secreted due to root rot, resin produced due to external damage to the trees, including test drillings performed prior to our study, was present in the collected data set. The labeling of the data for these two classification tasks differed significantly.

Finally, for each image, the disk was manually segmented from the original photograph and set over a black background, and the images were cropped closer to the outline of the disk. The resulting data set for the task of root rot detection comprised 192 disk images, of which 92 were "healthy" and 100 "rotten," forming an overall class distribution of 48% "healthy" vs. 52% "rotten." Examples of images in these two ground truth categories are presented in Figure 1(a). For the task of detecting resinous wood, the classes "contains extra resin" and "clean" comprised 103 and 89 images, respectively. This corresponds to a class distribution of 54% and 46% respectively. Examples of images in these two categories are presented in Figure 1(b). The annotated data sets are publicly available at http://dx.doi.org/10.5281/zenodo.5513331.

Splitting the data into training and test sets

A fundamental assumption in supervised machine learning is that the data being used is independent and identically distributed (i.i.d.). In our data, the i.i.d. assumptions are violated because there are dependencies between the disk images, as multiple stem end images were extracted from nearly every tree trunk. During the model selection process, if images from the same trunk are present in both the training and test data, i.e. the training and validation folds of the training set, the model is at risk of learning to recognize individual tree trunks instead of the target variable of interest, i.e. the presence of root rot or resin. Moreover, spreading the (relatively similar) images originating from a given tree trunk over the training and test sets would constitute a form of data leakage from the test set to the training set, which could lead to overly optimistic estimates of the model generalization error. To mitigate this issue, we took the following two actions.

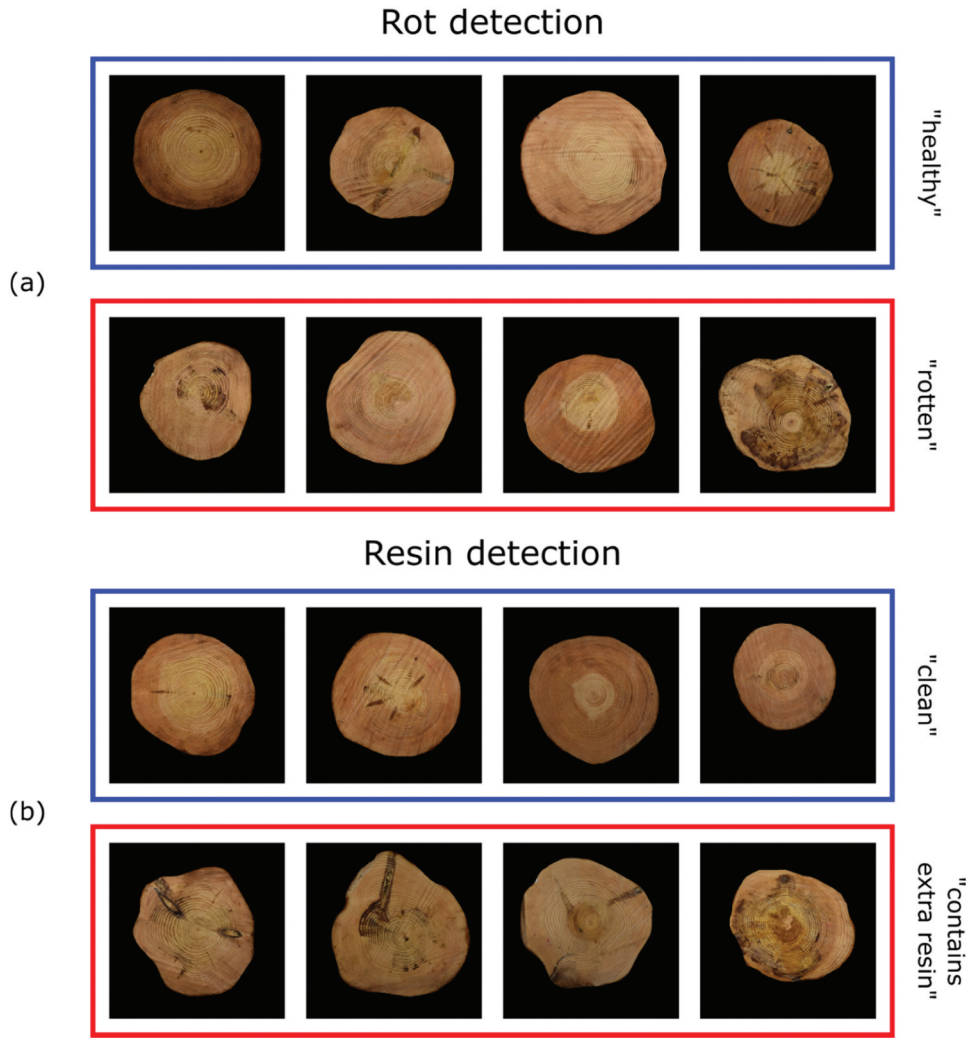

Figure 1. (a) Examples of images in each of the two categories for the classification task of detecting root rot disease in a stem end image of Scots pine. In the "healthy" category, the second image from the left contains decay that has not yet developed to degree four on the scale used here (see text for details). In the fourth image from the left, the resin is not due to root rot disease. In the "rotten" category, the decay is accompanied by resin in only two out of the four cases shown here. (b) Examples of images in each of the two categories for the classification task of detecting extra resin in a stem end image of Scots pine. Extra resin here means significant presence of resin outside branch knots.

First, disk images from a given trunk were only included in one data set at a time (training set, test set). To create the training set and test set for a given classification task, we repeated the random train/test split procedure until the (rot or resin) class distribution in both data sets was close to the overall class distribution in the data. For both rot detection and resin detection, the training set comprised 22 trunks, and the test set comprised the remaining 8 trunks. Details of the data sets for each of the tasks are given in Tables 1 and 2.

Second, in our model development procedure, we evaluated model candidates using leave-subject-out (LSO) cross-validation (Syrjälä et al. 2019). In this leave-one-out cross-validation variant, using the training set of 22 trunks, we trained the model on 21 trunks and evaluated the model on the remaining trunk. This process was repeated, with each of the trunks serving as the validation data at a time. Finally, the model performance metrics and loss were averaged over the 22 validation folds.

Table 1. Properties of the data sets used in developing and assessing our deep learning models for detecting root rot in stem end images of Scots pine. The overall fractions of healthy and rotten images over the entire data (192 images) are 0.48 and 0.52, respectively.

Property	Training set	Test set
Number of tree trunks	22	8
Number of healthy images	61	31
Number of rotten images	74	26
Total number of images	135	57
Fraction of healthy images	0.45	0.54
Fraction of rotten images	0.55	0.46

Pre-processing the images via a local binary pattern texture operator

In addition to developing rot detection models for regular RGB images of stem ends, we investigated the effect of pre-processing the images via a classical local binary pattern (LBP) texture operator (Ojala et al. 2002) prior to model development. The motivation behind this approach was the

Table 2. Properties of the data sets used in developing and assessing our deep learning models for detecting extra resin in stem end images of Scots pine. The overall fractions of clean and resinous images over the entire data (192 images) are 0.46 and 0.54, respectively.

Property	Training set	Test set
Number of tree trunks	22	8
Number of clean images	68	21
Number of resinous images	80	23
Total number of images	148	44
Fraction of clean images	0.46	0.48
Fraction of resinous images	0.54	0.52

idea that rotten areas in a stem end might systematically exhibit certain "textures," and extracting these from the image via the application of an LBP operator might make the classification task easier. We considered three variants of the LBP operator, as implemented in the scikit-image implementation of LBP (van der Walt et al. 2014): "default," "ror," and "uniform." We chose this set of LBP variants because it provided us with a reasonably comprehensive selection of different LBP approaches.

In each of these three LBP methods, the local texture around a given pixel in a grayscale image is quantified by a scalar number. In the "default" method, each pixel value p in a circularly symmetric local neighborhood of pixels is compared to the value c of the central pixel. The results of these comparisons are translated into 1's ($p - c \geq 0$) and 0's ($p - c < 0$), and the result is read out as a binary number which then quantifies the texture at the central pixel. In the "ror" method, rotational invariance is achieved by defining the LBP result at the central pixel as follows. One takes the "default" result for the binary number, and then performs bitwise right shifts on the number until the minimum value is obtained. Finally, in the "uniform" method, the focus is on so-called uniform patterns, which correspond to "default" binary numbers with at most two shifts between 0 and 1 when moving from one bit to the next. The texture is quantified at the central pixel as the sum of the results of the comparisons between neighborhood pixels and the central pixel, each comparison translated into 1 or 0 as in the "default" method. For a more detailed description of these three LBP methods, we refer the reader to Sections 2.1, 2.2, and 2.3 respectively of Ojala et al. (2002).

For each original RGB image, we first converted the image to grayscale. We then applied the LBP operator to the grayscale image. To obtain pixel values within the range [0, 255] for the LBP-filtered image, we then replaced each pixel value v_i with (Equation 1)

$$255 \times \left(v_i - \min_j \{v_j\} \right) / \left(\max_j v_j - \min_j \{v_j\} \right) \quad (1)$$

Furthermore, as each CNN here expected an image with three channels as input, we duplicated the single LBP channel to create an image with all three channels containing the same LBP pixel data. The pre-processing described here was done for each of the three LBP methods offline and prior to model development, for the same training set and test set as was used for rot detection from regular RGB images. An example of an RGB stem end image and the same image first converted into grayscale and then processed with each variant of the LBP operator considered here is presented in Figure 2. We used Python 3 for the LBP pre-processing and the development of all deep learning models in this study.

Developing the deep learning models

Outline of the model development procedure

In this study, we trained deep learning models for the following three tasks: rot detection using RGB images; rot detection using LBP-filtered RGB images; and resin detection using RGB images. As our data set was small for a deep learning task, we employed transfer learning (Yosinski et al. 2014). Here, using TensorFlow (Abadi et al. 2016) through the Keras framework (Chollet et al. 2015), we harnessed existing, general-purpose CNNs pre-trained on large image sets as feature extractors upon which we trained classifier layers specifically tuned for each task.

For our model's feature extractor component, we considered a range of CNNs of different architectures. Our candidate CNNs were EfficientNetB4 (Tan and Le 2019), MobileNet V2 (Sandler et al. 2018), Xception (Chollet 2017), ResNet50v2 (He et al. 2016), NASNetMobile (Zoph et al. 2018), VGG-19 (Simonyan and Zisserman 2014), and DenseNet201 (Huang et al. 2017). All are deep CNNs pre-trained for classification on a diverse subset of ImageNet data consisting of over one million annotated images in 1000 different categories (Russakovsky et al. 2015). The candidate CNNs all perform well on the ImageNet data, indicating that they are good general-purpose feature extractors with potential for transferability to a wide range of image recognition tasks. For details of these CNNs, we refer the interested reader to the given references.

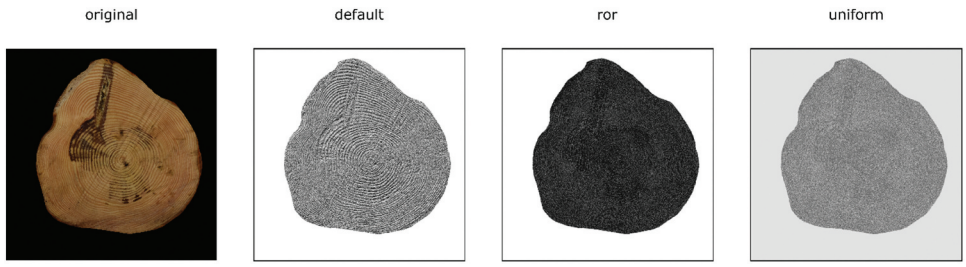

Figure 2. Example of a log end image of Scots pine before ("original") and after converting the image to grayscale and applying the LBP operator to the image. The three different LBP methods considered here were "default," "ror," and "uniform" (see text for details).

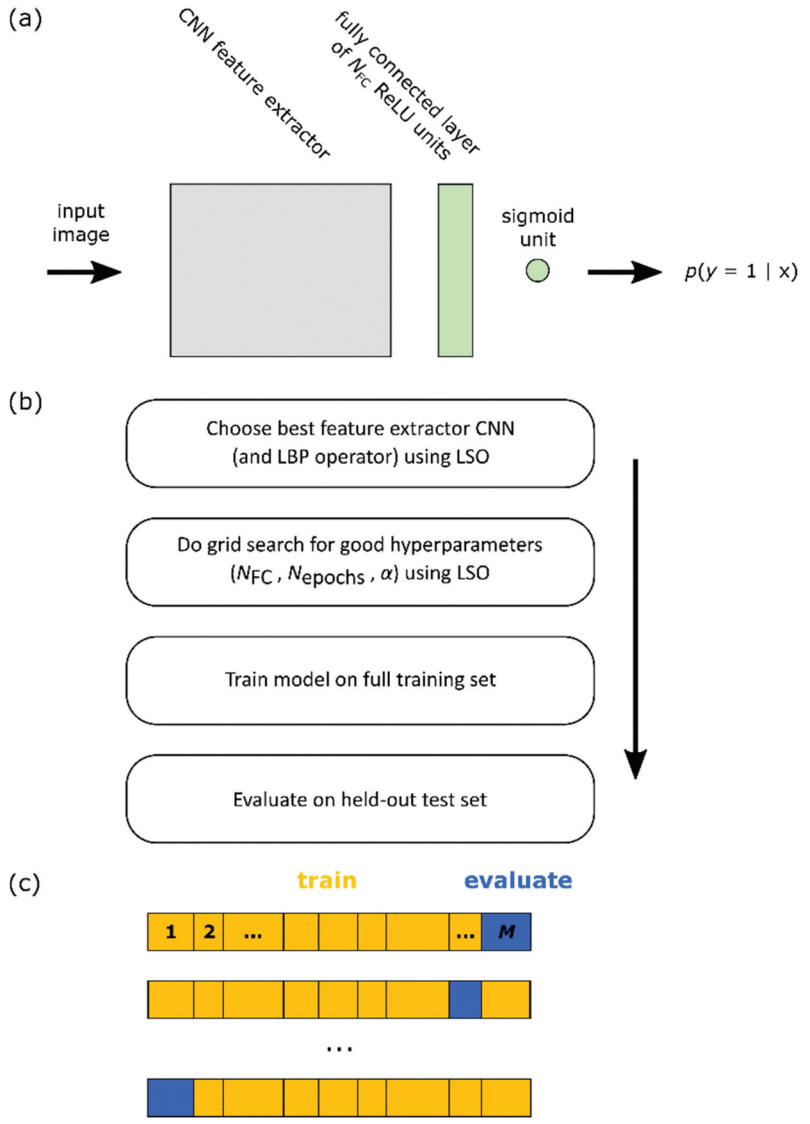

Figure 3. (a) Schematic view of the structure of the classification models developed in this study. (b) The workflow of model development. (c) Leave subject out (LSO) cross-validation as implemented in this study. The index 1, 2, . . . denotes all the images from trunk 1, all the images from trunk 2, etc. For a training set of M trunks, the model is trained on *M*-1 trunks (orange) and validated on the remaining trunk (blue). This is repeated until each trunk has served once as the validation data. The loss and performance metrics are then averaged over the M validation folds (see text for details).

The model structure we used for all three tasks is illustrated in Figure 3(a). The input image was fed to the feature extractor CNN, whose original classification layers had been removed and replaced with a 2D global average pooling layer. The CNN thus acted as a feature extractor outputting a 1D feature vector of "deep features" into which the content of the image were distilled. These features were then fed through a dropout layer (with a rate of 0.5 during training) to a fully connected layer of N_{FC} units with ReLU activation, which was followed by a single sigmoid unit. To classify a given image into the category $y = 1$, we adopted a threshold of 0.5 for the probability $p(y = 1 \mid x)$ output by the model.

For each of the three tasks considered here, we followed the same overall workflow in model development and evaluation (Figure 3(b)). First, using LSO cross-validation (Figure 3(c)), we selected the best-performing feature extractor CNN for the task. Again using LSO cross-validation, we then performed a search for optimal model hyperparameters. Next, we trained the model on the entire training set. Finally, we evaluated the model on the thus far held-out test set.

We used binary cross-entropy as the loss function and binary classification accuracy as the performance metric. In all training runs throughout the study, the batch size was 20, and gradient norms were clipped to 1.0. All network weights outside the batch normalization layers and the classification layers were kept frozen. To compute the validation accuracy in LSO cross-validation, we pooled the predictions and corresponding true labels over all the validation folds and computed the mean accuracy and standard error of the mean accuracy. To compute validation loss in the LSO cross-validation, we took the weighted average of the binary cross-entropy loss over the individual folds. During training, we performed augmentation of the training fold using random flips, shearing, translations, zooming, and rotations.

When training the models, we observed that for a given model configuration, the results often varied significantly

between different full cycles of LSO cross-validation. To evaluate a model configuration, both when choosing the best feature extractor (and LBP method) and in hyperparameter scanning, we therefore ran five full cycles of LSO cross-validation and took the mean of the accuracies and loss over these to represent the accuracy and loss of the model configuration in question.

In the following three sections, we describe the details of model development for each of the three tasks considered here.

Developing models for rot detection using RGB images

To first find the best-performing feature extractor CNN for this task, we set $N_{FC} = 0$, i.e. no fully connected layer between the pooling layer of the CNN and the sigmoid unit. Using each of the seven CNNs in turn, we trained the model for a period of $N_{epochs} = 100$ epochs at a learning rate of α = 1e-3 for the Adam optimizer (Kingma and Ba 2014) in stochastic gradient descent. All images were scaled to a size of 224 pixels by 224 pixels before inputting to the CNNs.

Based on these training runs, we chose the feature extractor CNN that gave the highest validation accuracy while exhibiting a reasonably low validation loss and smooth behavior of accuracy and loss as a function of training epoch. This CNN was ResNet50v2. Then, using ResNet50v2, we performed a grid search for optimal values of the following hyperparameters: α; N_{FC}; and N_{epochs}. We first varied α over 1e-2, 1e-3, 1e-4, N_{FC} over 0, 256, 2,048, and N_{epochs} over 25, 50, 100, 200, 300. From these runs, we chose the parameter combination that gave the highest validation accuracy, while exhibiting a reasonably low validation loss and smooth behavior of accuracy and loss as a function of the training epoch. As this set was α = 1e-3, N_{FC} = 0, $N_{epochs} = 300$, we performed another round of hyperparameter optimization, this time using α = 1e-3, $N_{FC} = 0$, and $N_{epochs} = 250$, 300, 350, 400, 500. Based on these results, the best setup for training our final model was ResNet50v2, α = 1e-3, $N_{FC} = 0$, and $N_{epochs} = 350$. The accuracy of the classifier using these hyperparameters, as averaged over the five full cycles of LSO cross-validation, was 76.7%.

We then trained the classifier on the entire training set and evaluated the resulting model on the thus far held-out test set. Because of significant variance in model performance between different randomly initialized training runs, we repeated this final training 50 times. To obtain baselines with which to compare our deep learning models, we used two simplistic models: one that 1) always predicted the dominant class in the training set, and one that 2) always predicted the dominant class in the test set.

Developing models for rot detection using RGB images pre-processed via LBP

In this task, in addition to finding the best-performing feature extractor CNN, we needed to find the best choice of LBP method to use. To accomplish this, we again considered the model structure with $N_{FC} = 0$. Then, using each of the seven CNNs in turn, and for each of them, using each of the three considered LBP methods in turn ("default," "ror," "uniform"), we trained the model for a period of $N_{epochs} = 1{,}000$ epochs at a learning rate of α = 1e-3. For the LBP methods, we used $P = 8$ and $R = 1$, where P is the number of points along the circularly symmetric neighborhood of radius R in pixels (see Ojala et al. 2002). Before being inputted to the CNNs, we scaled each LBP image to a size of 224 pixels by 224 pixels. The algorithm for performing the scaling proved significant to the visual appearance of the scaled images. Here, we chose the bilinear approach to obtain scaled LBP images similar in appearance to the full-sized LBP images.

From these runs, we chose the CNN and LBP combination that gave the highest validation accuracy, while still exhibiting a reasonably low validation loss and smooth behavior of accuracy and loss as a function of the training epoch. This combination was the Xception CNN and the "ror" variant of LBP.

Next, we explored different values of the LBP parameters P and R for the Xception + "ror" combination ($R = 2$, 3 with $P = 8$ R), repeating the LSO cycle twice for each combination of P and R. These tests showed that the original choice of $P = 8$, $R = 1$ resulted in the highest validation accuracy. We adopted this setup from there onward.

We then ran a hyperparameter search using α = 1e-2, 1e-3, 1e-4, $N_{FC} = 0$, 256, 2,048, and $N_{epochs} = 200$, 500, 1,000, 2,000. The parameter set that resulted in the highest validation accuracy, while still exhibiting a reasonably low loss and smooth behavior of loss and accuracy as a function of epoch was $N_{FC} = 2{,}048$, $N_{epochs} = 2{,}000$, and α = 1e-3. We therefore performed a further hyperparameter scan using α = 1e-3, $N_{FC} = 1{,}024$, 2,048, 4,096, and $N_{epochs} = 1{,}500$, 2,000, 2,500, 3,000. This provided no improvement on the results of the first hyperparameter scan, so we chose Xception + "ror" with $P = 8$, $R = 1$, $N_{FC} = 2{,}048$, $N_{epochs} = 2{,}000$, and α = 1e-3 for the final model setup. The accuracy of the classifier using these hyperparameters, as averaged over the five full cycles of LSO cross-validation, was 77.8%.

As in the case of using regular RGB images for rot detection, we repeated the final model training 50 times. We used the same simplistic models as a baseline for this task.

Developing models for resin detection using RGB images

To develop the CNN model for detecting extra resin in RGB images of the stem end, we used the same development and evaluation procedure as for the case of rot detection from RGB images, with the following exception: only a single round of hyperparameter scanning was necessary here. The best setup for training our final model was VGG-19, α = 1e-3, $N_{FC} = 256$, and $N_{epochs} = 100$. The accuracy of the classifier using these hyperparameters, as averaged over the five full cycles of LSO cross-validation, was 75%. We again repeated the final model training 50 times. Corresponding to the cases of the two previous sections, for comparing our deep learning models against, we used two simplistic models: one that 1) always predicted the dominant class in the training set, and one that 2) always predicted the dominant class in the test set.

Results

Automatic detection of root rot from RGB images

For the task of rot detection in regular RGB images of stem ends, the mean accuracy over the 50 individual final training runs was (64.0 ± 0.5)%, where the error is the standard error of the mean.

This is somewhat lower than the accuracy of 76.7% obtained at the end of model development using LSO, possibly indicating a slight overfitting of the model on the training set. In the following, we present detailed results for a single representative model from this ensemble, which we call "the final model."

The performance metrics for the final model, as computed on the test set, are summarized in Table 3. The overall classification accuracy on the test set was (63 ± 6)%, where the error is the standard error. The confusion matrix for this classifier is highly unsymmetrical (Table 4), which is reflected in the large difference between precision and recall. The model clearly favors "rotten" to "healthy" in its predictions, as evinced by the lack of false negatives and a fair amount of false positives on the test set. The F1 score is a reasonably good 0.71. In contrast, the area under the receiver operating characteristics curve (Figure 4) is very high (0.95). This may be due to the fact that as the model emphasizes "rotten," i.e. positive predictions over "healthy," i.e. negative ones, the true positive rate necessarily tends to be high.

The classification accuracy was (54 ± 7)% for a baseline model that always predicted the dominant class in the test set and (46 ± 7)% for one that predicted the dominant class in the training set. Overall, despite the overlap of the accuracy of the final model with the best baseline result, the final model appears to perform better than these baseline models on the test set, including an area of 0.95 under the ROC curve compared to the value of 0.5 given by the family of "no skill" models producing uniformly random predictions, to which our baseline models belong. Case examples of correct and incorrect predictions given by the final model on the test set are presented in Figure 5.

Automatic detection of root rot from RGB images pre-processed via LBP

For the task of rot detection in stem end RGB images pre-processed via LBP, the mean accuracy over the 50 final training runs was (70.1 ± 1.4)%. This is slightly lower than the accuracy of 77.8% obtained at the end of hyperparameter tuning using LSO cross-validation, which may indicate a small overfitting. As in the previous section, we present detailed results in the following for a single representative model from this ensemble, which we call "the final model."

The performance metrics for the final model, as computed on the test set, are summarized in Table 3. The overall classification accuracy on the test set was (70 ± 6)%. As for the case of rot detection directly from RGB images, the confusion matrix for this classifier is highly unsymmetrical (Table 4), which leads to a large difference between precision and recall. This model also clearly favors "rotten" to "healthy" in its predictions, as shown by the significant number of false positives and only a single false negative on the test set. The F1 score is a reasonably good 0.75. In contrast, the area under the receiver operating characteristics curve (Figure 4) is again very high (0.93), possibly for the reason suggested in the previous section.

With an accuracy of (70 ± 6)%, the final model is significantly better than the best baseline model, which had an accuracy of (54 ± 7)%. Examples of correct and incorrect predictions given by the final model on the test set are presented in Figure 6.

Automatic detection of resin from RGB images

For the task of detecting extra resin in RGB images of stem ends, the mean accuracy over the 50 final training runs was (79.4 ± 0.5)%. This is slightly higher than the accuracy of 75% found at the end of model development using LSO cross-validation, but the small difference may simply be explained by the small size of the test set. As in the previous two sections, we present detailed results in the following for a single representative model from this ensemble, which we call "the final model."

Table 3. Test set performance metrics for each of the classifier models developed in this study. The F1 score is the harmonic mean of the precision and recall. See Alpaydin (2016) for the definition of the rest of these metrics. For each case, the results are given for the representative final model (see text for details). The error in the accuracy is the standard error.

Metric	Rot detection from RGB images	Rot detection from RGB images pre-processed via LBP	Extra resin detection from RGB images
Accuracy	0.63 ± 0.06	0.70 ± 0.06	0.80 ± 0.06
Precision	0.55	0.61	0.73
Recall	1.0	0.96	0.96
F1 score	0.71	0.75	0.83
Area under ROC curve	0.95	0.93	0.87

Table 4. The confusion matrix on the test set for each of the classifier models developed in this study. The results are given for the representative final model for each case (see text for details).

		Ground truth	
		Healthy	Rotten
Model prediction, rot detection from RGB images	Healthy	10	0
	Rotten	21	26
		Healthy	Rotten
Model prediction, rot detection from RGB images pre-processed via LBP	Healthy	15	1
	Rotten	16	25
		Clean	Resinous
Model prediction, extra resin detection from RGB images	Healthy	13	1
	Rotten	8	22

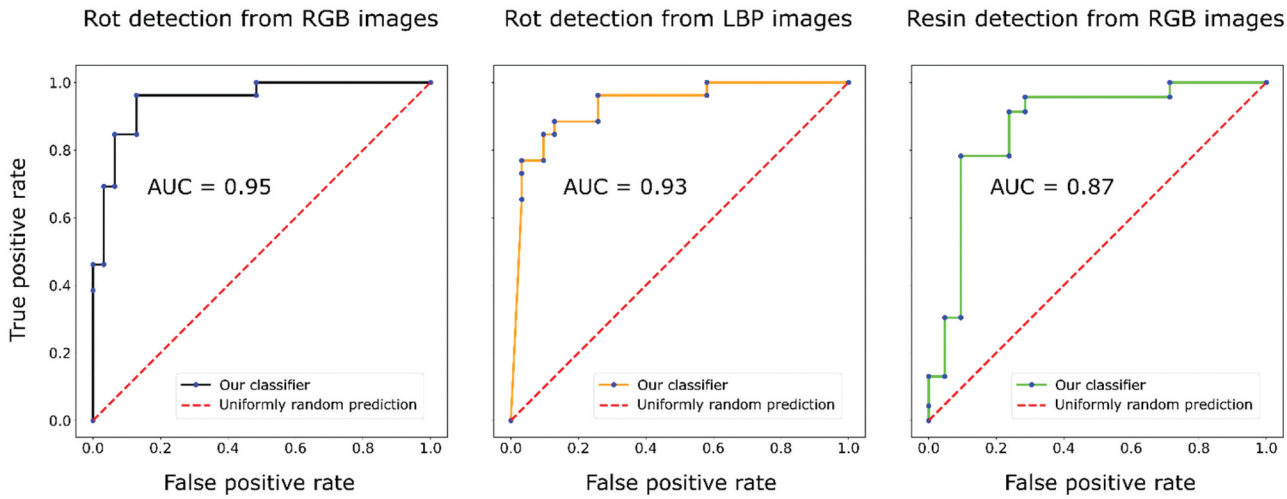

Figure 4. Receiver operating characteristics curve on the test set for each of the three classifiers developed in this study. For each case, the classifier in question is the representative final model (see text for details). The area-under-the-curve (AUC) value is reported in the plot for each classifier. The models represented by the red dashed line produce uniformly random predictions for p(y = 1 | x).

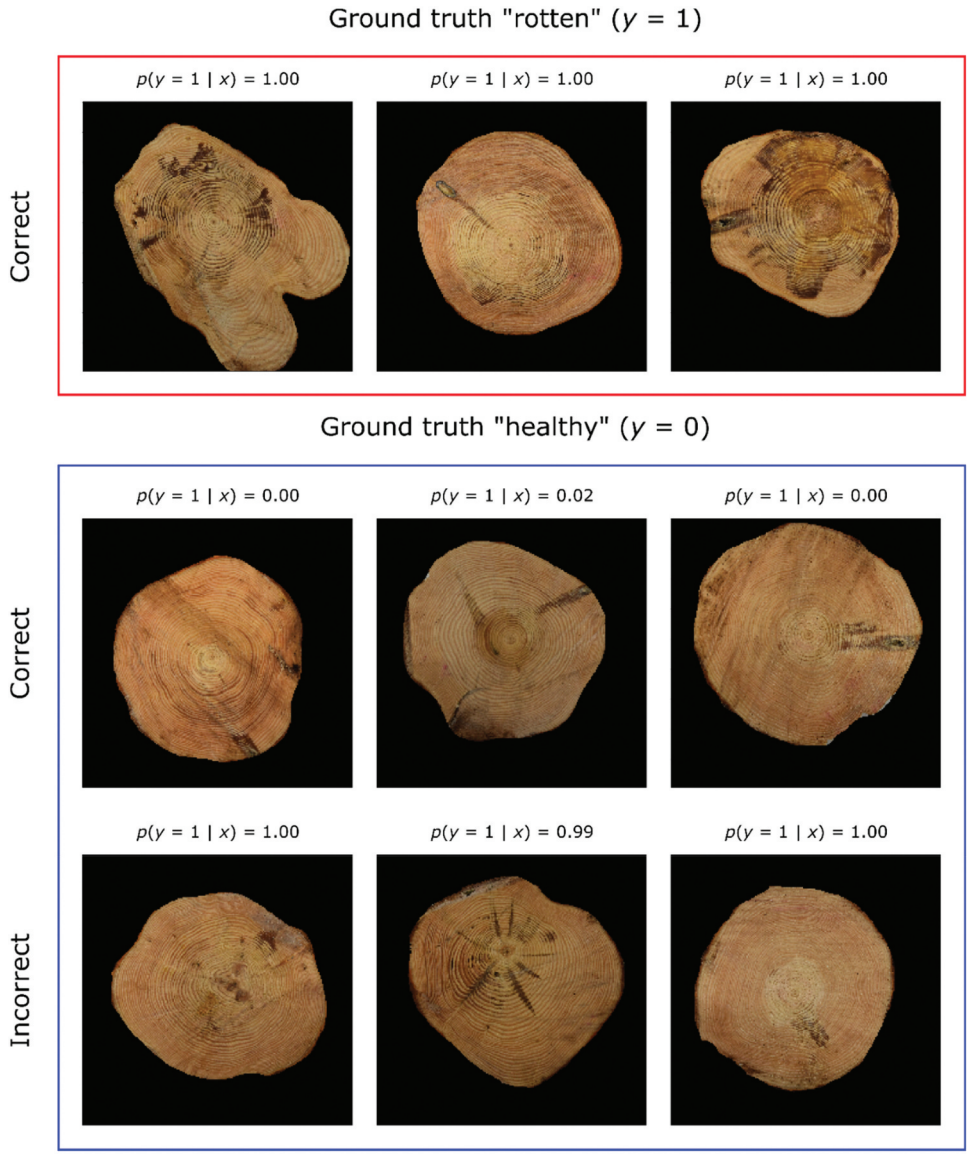

Figure 5. Examples of predictions made on the test set by the final representative classifier model (see text for details) for detecting root rot in RGB stem end images of Scots pine. No incorrect predictions, i.e. false negatives, were produced by the model in the rotten ground-truth category (y = 1).

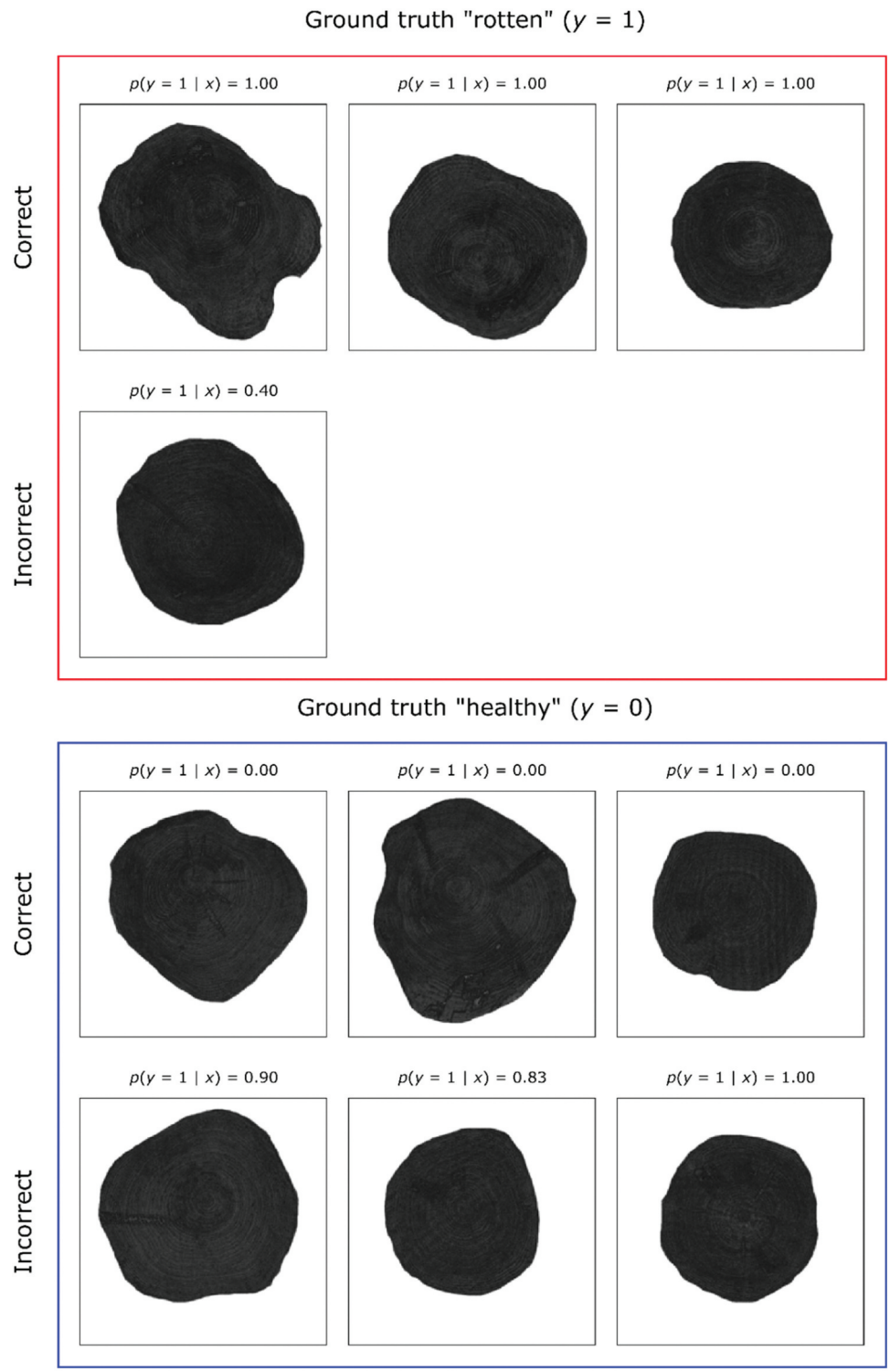

Figure 6. Examples of predictions made on the test set by the final representative classifier model (see text for details) for detecting root rot in LBP-processed stem end images of Scots pine. In the rotten ground-truth category, only one incorrect prediction, i.e. false negative, was produced by the model.

The performance metrics for the final model, as computed on the test set, are summarized in Table 3. The overall classification accuracy on the test set was (80 ± 6)%. The confusion matrix for this classifier is more symmetrical than for the models of the two preceding sections (Table 4). However, this model also clearly favors positive to negative in its predictions, as shown by a fair number of false positives and only a single false negative on the test set. The F1 score is 0.83, higher than for the previous two cases. The area under the receiver operating characteristics curve (Figure 4) is again very high (0.87), possibly for the same reason as suggested for the case of rot detection from RGB images.

For a baseline model that always predicts the class "contains extra resin," which is the dominant class in both the

training and the test set, the accuracy on the test set is (52 ± 8)%. With an accuracy of (80 ± 6)%, our final model is significantly better than this baseline. Examples of correct and incorrect predictions given by the final model on the test set are presented in Figure 7.

Discussion

The purpose of this study was to develop CNN-based deep learning methods to automatically detect root rot disease and resin patches in stem end images of Scots pine. Our results show that the detection of the presence of severe root rot is

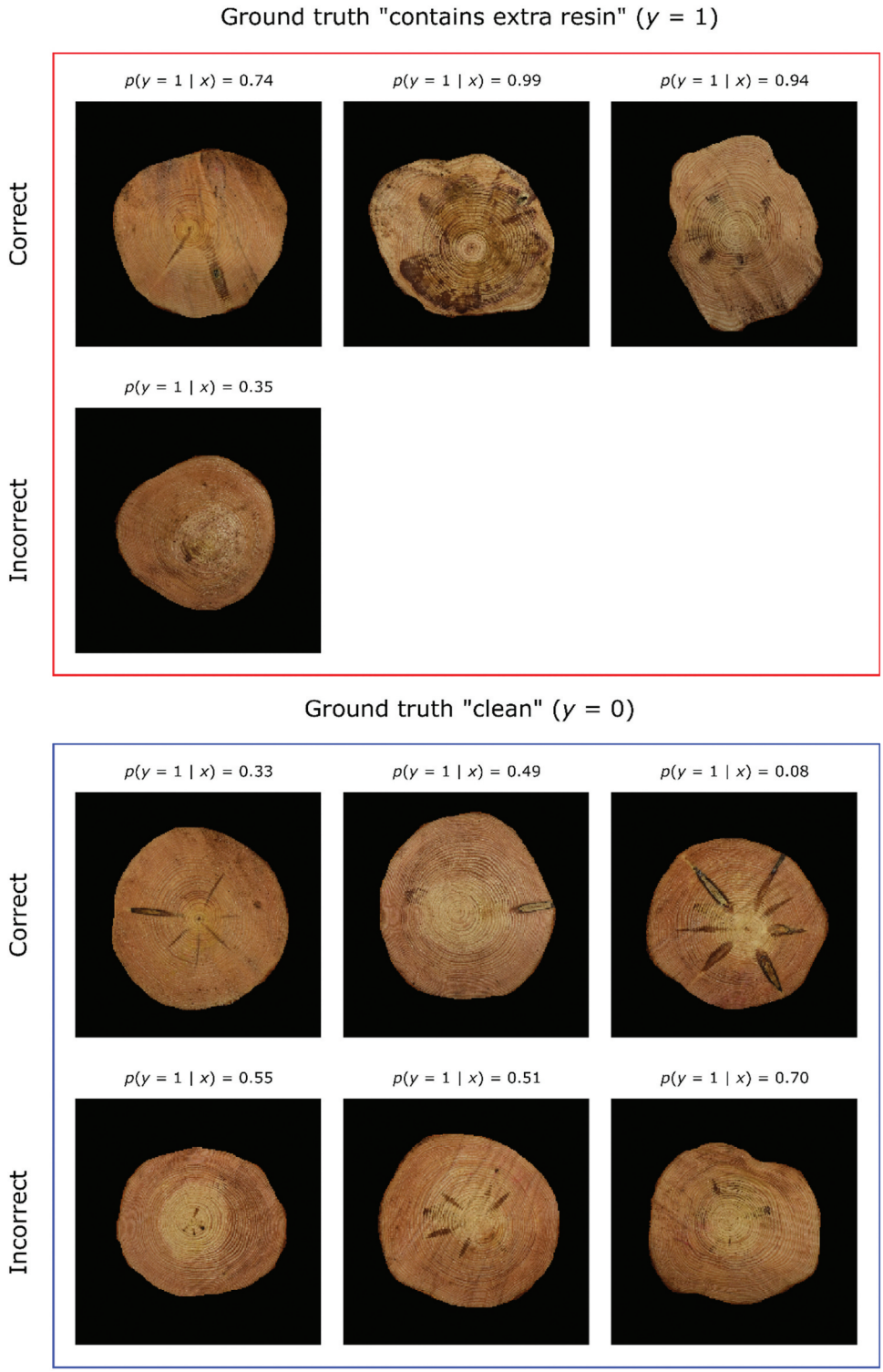

Figure 7. Examples of predictions made on the test set by the final representative classifier model (see text for details) for detecting extra resin in stem end RGB images of Scots pine. In the rotten ground-truth category, only one incorrect prediction, i.e. false negative, was produced by the model.

possible in Scots pine, although the achieved classification accuracy of (63 ± 6)% remains modest. However, when the stem end images are pre-processed via the classical LBP texture operator prior to model development, the final classification accuracy rises to (70 ± 6)%. It therefore appears that using the chosen LBP operator may help bring out textures indicative of the presence or absence of root rot disease in the tree. This then makes it easier to learn the desired mapping using a deep learning classifier.

The automatic detection of resin in a stem end image of Scots pine appears a simpler task than root rot detection. In resin detection, we find a final classification accuracy as high as (80 ± 6)% using regular RGB images. It is foreseeable that pre-filtering the images via LBP could also improve the result of the resin detection task. Testing this prediction is a subject for future study.

Previously published studies of automatic detection of root rot in cross-sectional images of trees focus on Norway spruce. Using a random forest classifier applied to a set of 309 stump images at a spruce-dominated site, segmented either automatically or manually from UAV photographs, Puliti et al. (2018) achieved an overall classification accuracy of 78.3% for the presence of root and butt rot using out-of-bag error estimation. In their study, there was a delay of approximately two months between harvesting the trees and imaging the stumps. Using a 521-image subset of a data set of 1,000 images of Norway spruce stumps acquired on foot and segmented manually, Ostovar et al. (2019) developed a model consisting of the VGG-19 CNN and dedicated classification layers, reaching a binary classification accuracy of 91%. In their study, the convolutional layers of the CNN were also trained, i.e. fine-tuned to the task. The images were acquired within a few days of the harvesting operation. Nowell (2019) analyzed the full 1,000-image data set underlying the study of Ostovar et al. (2019), using manual segmentation of stumps, and found an F1 score of 97.1% for classification using the fine-tuned DenseNet201 CNN as the feature extractor.

Compared with these earlier results regarding the automatic detection of root rot in cross-sectional images of trees, the classification accuracy of (70 ± 6)% and F1 score of 75% given by our LBP + CNN model appears somewhat modest. A reason for this difference may be that root rot in Scots pine is an inherently more subtle phenomenon than in the case of Norway spruce. In Norway spruce, the decay column is often very prominent and rises several meters from the ground upward into the stem. In contrast, root rot in Scots pine rarely rises above a few dozen centimeters into the stem, and the decay is typically more difficult to recognize visually.

Another factor that probably contributes to the difference is that our data set of 192 images is smaller than the data sets employed in these earlier studies focusing on Norway spruce and, in particular, can be considered small for learning the somewhat diffuse mapping between images and the presence of root rot disease discussed above. However, in Puliti et al. (2018), Ostovar et al. (2019), and Nowell (2019), the images were collected from various different angles and distances from the tree stumps at different levels of lighting, and occlusions by, e.g. needles and branches were allowed, which probably made the root rot classification task more challenging. Our setup corresponds perhaps most closely to a situation in which the camera in the harvesting operation is inside the harvester head, and dust and dirt, etc. are, e.g. blown away with air pressure prior to imaging.

To develop a root rot classifier for stem end images of Scots pine, using a larger number of images than was available in the present study would probably produce better results, possibly sufficiently good for implementation in real-life operations. A larger data set would also allow the entire CNN network to be fine-tuned to the task at hand, a procedure which we avoided here due to early experimentation indicating a high risk of overfitting the model on the small training set. We also note that when labeling our data, depending on whether a low-lying disk in the stem was deemed to display root rot disease, resin found higher up in that stem was respectively attributed either to the disease or another cause. In light of this ambiguity inherent in the visual disease symptoms, learning a *perfect* mapping from stem end images to the presence or absence of root rot may be impossible. Leveraging the electromagnetic spectrum beyond the visual range, e.g. the near-infrared (Liaghat et al. 2014; Abdulridha et al. 2016), might present a route toward higher detection accuracy for the disease.

Models that nearly attain or even surpass human accuracy in visual detection tasks regarding the stem during CTL harvesting operations could increase the cost-effectiveness of forest operations. In addition, the automatic detection of root rot and other abnormal properties of harvested stems and subsequent decision support regarding bucking optimization would act to reduce the currently high mental workload of harvester operators.

Regarding the required model performance level for real-life operations, from the perspective of the wood purchaser, avoiding false negatives in the detection of decay is probably a priority. This is because rotten wood that ends up at the sawmill (false negative) implies a higher financial loss than healthy wood taken for rotten and transported for pulp production (false positive). Decayed wood at the sawmill also constitutes a form of resource loss, the log being more likely to be used for burning than for sawed timber. On the other hand, the seller of the wood should desire the fewest possible false positives, as a healthy log mistakenly considered rotten implies a smaller purchase price for the wood. These conflicting perspectives underline the need for very high accuracy in root rot detection in real-world forest operations.

Finally, to improve the results of root rot detection obtained in this study, it may be beneficial to "extend" the dataset by operating on close random crops of the stem end images instead of on one full cross-sectional image at a time. This would also increase the resolution of the finer details of the stem end, which could help in learning the desired mapping. However, the considerations presented above regarding the ambiguous character of the mapping from images to the presence of disease probably set a limit for the attainable accuracy.

Conclusions

Our results indicate that automatically detecting root rot disease in regular RGB images of Scots pine stem ends is challenging. Pre-filtering the images using a classical texture operator appears to enhance the performance of the deep learning models, raising classification accuracy to a more promising level. Despite this, the ambiguous nature of the correspondence between the visual appearance of a stem end and the presence or absence of the disease may pose a limit for the maximum attainable accuracy in this task using regular photography. Access to wavelengths beyond the visual spectrum may provide a solution to this problem. Automatic detection of resinous wood appears a simpler and more feasible task than root rot disease detection. The automatic analysis of wood quality at harvesting has the potential to improve the cost-effectiveness of harvesting and decrease the mental workload of harvester operators. However, very low error rates will probably be required to satisfy all the parties in the wood procurement process.

Acknowledgements

The authors wish to thank Timo Pitkänen and Timo Siitonen for the field work involved in the data collection for this study. In addition, EH wishes to acknowledge CSC – IT Center for Science, Finland for computational resources.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This research was supported by the Academy of Finland Flagship "Forest-Human-Machine Interplay – Building Resilience, Redefining Value Networks and Enabling Meaningful Experiences" (funding decision 337655), and by the TyviTuho project funded by the Ministry of Agriculture and Forestry of Finland through the Catch the carbon research and innovation program (funding decision VN/5206/2021).

ORCID

Eero Holmström http://orcid.org/0000-0002-4866-3730

References

Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado GS, Davis A, Dean J, Devin M, et al. 2016. TensorFlow: large-scale machine learning on heterogeneous distributed systems. arXivorg. https://arxiv.org/abs/1603.04467.

Abdulridha J, Ehsani R, de Castro A. 2016. Detection and differentiation between Laurel Wilt disease, phytophthora disease, and salinity damage using a hyperspectral sensing technique. Agriculture. 6(4):56. doi: 10.3390/agriculture6040056.

Alpaydin E. 2016. Introduction to machine learning. Cambridge, MA, USA: MIT Press.

Burdekin DA. 1972. A study of losses in scots pine caused by *fomes annosus*. Forestry. 45(2):189–196. doi: 10.1093/forestry/45.2.189.

Chollet F. 2015. Keras. https://keras.io.

Chollet F 2017. Xception: deep learning with depthwise separable convolutions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; Honolulu, USA. p. 1251–1258.

Fabijańska A, Danek M. 2018. DeepDendro – a tree rings detector based on a deep convolutional neural network. Comp Electr Agr. 150:353–363. doi: 10.1016/j.compag.2018.05.005.

Garbelotto M, Gonthier P. 2013. Biology, epidemiology, and control of *heterobasidion* species worldwide. Ann Rev Phyt. 51(1):39–59. doi: 10.1146/annurev-phyto-082712-102225.

Goodfellow I, Bengio Y, Courville A. 2016. Deep learning. Cambridge: MIT Press. https://www.deeplearningbook.org/.

Greig BJW. 1995. Butt-rot of scots pine in Thetford Forest caused by *heterobasidion annosum*: a local phenomenon. For Path. 25(2):95–99. doi: 10.1111/j.1439-0329.1995.tb00323.x.

He K, Zhang X, Ren S, Sun J. 2016. Identity mappings in deep residual networks. Computer Vision - ECCV; Amsterdam, The Netherlands. p. 630–645.

Holmström E, Raatevaara A, Pohjankukka J, Korpunen H, Uusitalo J. 2023. Tree log identification using convolutional neural networks. Smart Agric Technol. 4:100201. doi: 10.1016/j.atech.2023.100201.

Huang G, Liu Z, van der Maaten L, Weinberger KQ. 2017. Densely connected convolutional networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; Honolulu, USA. p. 4700–4708.

Kingma DP, Ba J. 2014. Adam: a method for stochastic optimization. arXivorg. https://arxiv.org/abs/1412.6980.

Laine L. 1976. The occurrence of *Heterobasidion annosum* (Fr.) Bref. In woody plants in Finland. Metsant Julk. 90(3).

Liaghat S, Ehsani R, Mansor S, Shafri HZM, Meon S, Sankaran S, Azam SHMN. 2014. Early detection of basal stem rot disease (Ganoderma) in oil palms based on hyperspectral reflectance data using pattern recognition algorithms. Int J Rem Sens. 35 (10):3427–3439. doi: 10.1080/01431161.2014.903353.

Müller MM, Sievänen R, Beuker E, Meesenburg H, Kuuskeri J, Hamberg L, Korhonen K, Sturrock RN. 2014. Predicting the activity of *Heterobasidion parviporum* on Norway spruce in warming climate from its respiration rate at different temperatures. For Path. 44 (4):325–336. doi: 10.1111/efp.12104.

Mäkinen H, Korpunen H, Raatevaara A, Heikkinen J, Alatalo J, Uusitalo J. 2019. Predicting knottiness of Scots pine stems for quality bucking. Eur J W W Prod. 78(1):143–150. doi: 10.1007/s00107-019-01476-x.

Nowell TC. 2019. Detection and quantification of rot in harvested trees using convolutional neural networks. Norway: Norwegian University of Life Sciences, Ås.

Ojala T, Pietikainen M, Maenpaa T. 2002. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans Pat An Mach Intel. 24(7):971–987. doi: 10.1109/TPAMI.2002.1017623.

Ostovar A, Talbot B, Puliti S, Astrup R, Ringdahl O. 2019. Detection and classification of root and butt-rot (RBR) in stumps of Norway spruce using RGB images and machine learning. Sensors. 19(7):1579. doi: 10.3390/s19071579.

Pitkänen TP, Piri T, Lehtonen A, Peltoniemi M. 2021. Detecting structural changes induced by heterobasidion root rot on scots pines using terrestrial laser scanning. For Ecol Manag. 492:119239. doi: 10.1016/j.foreco.2021.119239.

Puliti S, Talbot B, Astrup R. 2018. Tree-stump detection, segmentation, classification, and measurement using unmanned aerial vehicle (UAV) imagery. Forests. 9(3):102. doi: 10.3390/f9030102.

Robert M, Dallaire P, Giguère P. 2020. Tree bark re-identification using a deep-learning feature descriptor. 17th Conference on Computer and Robot Vision (CRV); Ottawa, Canada.

Ruotsalainen S. 2014. Increased forest production through forest tree breeding. Scand J For Res. 29(4):333–344. doi: 10.1080/02827581.2014.926100.

Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, et al. 2015. ImageNet Large Scale Visual Recognition Challenge. Int J Comp V. 115(3):211–252. doi:10.1007/s11263-015-0816-y.

Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L-C. 2018. MobileNetV2: Inverted Residuals and Linear Bottlenecks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; Salt Lake City, USA. p. 4510–4520.

Seidl R, Thom D, Kautz M, Martin-Benito D, Peltoniemi M, Vacchiano G, Wild J, Ascoli D, Petr M, Honkaniemi J, et al. 2017. Forest disturbances under climate change. Nat Clim Chan. 7(6):395–402. doi:10.1038/nclimate3303.

Simonyan K, Zisserman A. 2014. Very deep convolutional networks for large-scale image recognition. arXivorg. https://arxiv.org/abs/1409.1556.

Syrjälä E, Jiang M, Pahikkala T, Salanterä S, Liljeberg P. 2019. Skin Conductance Response to Gradual-Increasing Experimental Pain. 41st Annual International Conference; Berlin.

Tan M, Le Q. 2019. EfficientNet: rethinking model scaling for convolutional neural networks. PMLR. 97:6105–6114.

van der Walt S, Schönberger JL, Nunez-Iglesias J, Boulogne F, Warner JD, Yager N, Gouillart E, Yu T. 2014. Scikit-image: image processing in python. PeerJ. 2:e453. doi: 10.7717/peerj.453.

Vihlman M, Kulovesi J, Visala A. 2019. Tree log identity matching using convolutional correlation networks. International Conference on Digital Image Computing: Techniques and Applications; Perth, Australia.

Wang L, Zhang J, Drobyshev I, Cleary M, Rönnberg J. 2014. Incidence and impact of root infection by *Heterobasidion* spp., and the justification for preventative silvicultural measures on scots pine trees: a case study in southern Sweden. For Ecol And Manag. 315:153–159. doi: 10.1016/j.foreco.2013.12.023.

Yosinski J, Clune J, Bengio Y, Lipson H. 2014. How transferable are features in deep neural networks?. Advances in Neural Information Processing Systems; Montréal, Canada.

Zoph B, Vasudevan V, Shlens J, Le QV. 2018. Learning transferable architectures for scalable image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; Salt Lake City, USA. p. 8697–8710.