

1 **Predicting tree failure likelihood for utility risk mitigation via artificial**
2 **intelligence**

3 Jimi Oke¹, Nasko Apostolov¹, Ryan Suttle², Sanjay Arwade¹, and Brian Kane²

4 ¹Department of Civil and Environmental Engineering, University of Massachusetts Amherst, MA
5 01003, USA

6 ²Department of Environmental Conservation, University of Massachusetts Amherst, MA 01003,
7 USA

8 **ABSTRACT**

9 Critical to the resilience of utility power lines, tree failure assessments have historically been
10 performed via manual and visual inspections. In this paper, we develop a convolutional neural
11 network (CNN) to predict tree failure likelihood categories (*Probable*, *Possible*, *Improbable*)
12 under four classification scenarios. Starting with an original set of 505 expert-labeled images, we
13 performed preprocessing and augmentation tasks to increase the number of samples by a factor of 5.
14 We optimized several hyperparameters in the CNN and then trained and assessed its performance via
15 cross-validation for three different image resolutions under all four scenarios. The CNN produced
16 a validation accuracy of at least 0.94 ($\hat{\sigma} = 0.1$) in the best-performing, yet hypothetical scenario
17 as it excludes one of the categories. The second-best performing scenario, which includes all
18 categories and therefore more practical, resulted in a validation accuracy of 0.92 ($\hat{\sigma} = 0.1$). Thus,
19 via this novel framework, we demonstrate the potential of artificial intelligence to automate and
20 consequently reduce the costs of tree failure likelihood assessments, thereby promoting sustainable
21 infrastructure.

22 **INTRODUCTION**

23 Despite extensive efforts by utilities to prevent them, contacts between tree parts and power
24 lines cause outages that annually result in tens of billions of dollars in economic costs throughout

25 the United States. Presently, the identification of potential contact between trees and power lines
26 is labor intensive and time-consuming. This paper describes an artificial intelligence and machine
27 learning approach that automatically classifies trees, using only a single photograph and with a high
28 degree of accuracy, into categories used by utility arborists to describe the likelihood of tree failure:
29 probable, possible, and improbable. This preliminary study demonstrates the possible efficacy of
30 AI approaches to tree risk assessment and, following further development of the approach, has the
31 potential to reduce power outages and utility costs by allowing utilities to more effectively target
32 their pruning and mitigation efforts.

33 Contact between tree parts and power lines can take three forms: tree branches can grow into
34 lines; branches can fail and fall onto lines; whole-tree failure can occur due to uprooting or trunk
35 failure. A study in Connecticut, USA provides some context for the amount of economic disruption,
36 documenting annual disruptions of \$8.3 billion between 2005 and 2015 ([Graziano et al. 2020](#)). That
37 extremely high cost occurred despite extensive efforts on the part of utilities to mitigate conflicts
38 between trees and power lines through active and aggressive pruning programs that, on their own
39 cost billions of dollars annually ([Guggenmoos 2003](#)).

40 Despite its high cost, pruning trees to maintain clearance from power lines is an effective way
41 to reduce outages due to so-called “preventable” contacts between trees and power lines. For
42 example, in Massachusetts, USA, where tree failure was responsible for 40% of preventable tree-
43 caused outages, pruning was able to improve reliability by 20% to 30% ([Simpson and Van Bossuyt
1996](#)); similar results were found in a study conducted in Connecticut ([Parent et al. 2019](#)). The
44 efficacy of pruning has also been shown in a study of two states in the Gulf Coast region of the USA
45 that showed wind-induced power outage prediction models becoming less uncertain when pruning
46 was included in the model ([Nateghi et al. 2014](#)).

48 Even effective pruning cannot, however, completely eliminate tree-caused outages. Failure of
49 trees outside the right-of-way can still impact the lines and cause outages ([Guggenmoos 2003](#)). The
50 proportion of outages caused by failure of trees outside the right-of-way has not been rigorously
51 quantified. [Guggenmoos \(2011\)](#) estimated that 95% of tree-caused outages in the Pacific Northwest

52 region of the USA, were due to tree failure, and [Wismer \(2018\)](#) reported approximately 25% of
53 interruptions in Illinois, USA, were caused by trees that uprooted or broke in the stem.

54 Predicting the likelihood of failure is an inexact science, but tree risk assessment best manage-
55 ment practices have been developed ([Smiley et al. 2017](#); [Goodfellow 2020](#)). Estimating tree risk
56 includes assessing the likelihood of tree failure, the likelihood of impact of the failed tree (or tree
57 part) on a target, and the severity of consequences of the impact. The likelihood of failure depends
58 on the anticipated loads on the tree and its load-bearing capacity. The likelihood of impact depends
59 on proximity to the target (the lines, poles, and other hardware—“infrastructure”—in the case of
60 utility tree risk assessment), the target’s occupancy rate (which is constant for utility lines) and
61 whether the target is sheltered, for example by neighboring trees. Severity of consequences depends
62 on the damage done to the infrastructure—which, in turn, is partially related to the size of the tree or
63 tree part that fails, and how much momentum it has when it impacts the infrastructure—and, more
64 importantly in some cases, the economic costs and disruption associated with electrical outages.

65 Individual tree risk assessment can be costly because of the time it requires. In some situations,
66 a less time-consuming assessment may be justified to reduce costs, i.e. a “Level 1” assessment
67 ([Smiley et al. 2017](#)). Studies in Rhode Island, USA ([Rooney et al. 2005](#)) and Florida, USA
68 ([Koeser et al. 2016](#)) have shown that, compared to more time-consuming risk assessments, Level
69 1 risk assessments successfully identified trees with a higher degree of risk—precisely the trees that
70 arborists prioritize for risk mitigation. The utility of Level 1 assessments demonstrated in these
71 studies suggests that artificial intelligence (AI) tools may be an effective way to reduce the cost of
72 tree risk assessment while still identifying high risk trees.

73 The method described in the paper uses convolutional neural networks (CNN) to classify images
74 of trees among three categories of failure likelihood: probable, possible, and improbable. The data
75 used for training, testing and illustration of the method consists of 505 tree images that have
76 been classified by the authors according to best management practices used by utility arborists
77 ([Goodfellow 2020](#)).

78 The remainder of the paper provides a brief history and background of AI and its use in

79 infrastructure risk assessment and tree identification (section 2); describes the methods used to
80 train and validate a novel CNN to categorize likelihood of tree failure (section 3); and presents and
81 discusses the output of the novel CNN (sections 4 and 5). The goal is to further demonstrate an
82 innovative automated approach to tree risk assessment using an AI tool that can be readily deployed
83 for use in various locations and also continually improved through subsequent training on new
84 datasets.

85 **BACKGROUND**

86 AI-based image analysis is relatively widely used, even in engineering applications, such as
87 earthquake risk assessment ([Jiao and Alavi 2020](#); [Salehi and Burgueño 2018](#)) and structural health
88 monitoring ([Spencer et al. 2019](#); [Wang et al. 2019](#)). Neural networks, which comprise a major
89 category of AI frameworks, have been widely applied in the field of earthquake risk assessment
90 (an excellent review is provided by [Xie et al. \(2020\)](#)), but the authors are not aware of attempts
91 to operate directly on, for example, building images in the absence of technical structural data to
92 predict seismic risk. Neural networks have also been used to interrogate remote sensing data of
93 the landscape to assess landslide risk ([Su et al. 2020](#)). A few recent efforts have demonstrated
94 the potential for AI-based tree recognition from drone imagery ([dos Santos et al. 2019](#); [Egli and](#)
95 [Höpke 2020](#)). Furthermore, an application of a convolutional neural network (CNN) to tree species
96 identification using was recently demonstrated by [Fricker et al. \(2019\)](#). Yet, AI has yet to be applied
97 to the problem of tree-utility line risk assessment—one that is complicated by the very large number
98 of tree species to be considered, seasonal variation in tree appearance and associated risk and local
99 meteorological conditions.

100 The groundbreaking study of [Hubel and Wiesel \(1959\)](#) showed that visual perception in cats
101 was a result of the activation or inhibition of groups of cells in the visual cortex known as “receptive
102 fields.” Further, they attempted to map the cortical architecture in cats and monkeys ([Hubel and](#)
103 [Wiesel 1962](#); [Hubel and Wiesel 1965](#); [Hubel and Wiesel 1968](#)). Subsequent attempts were then
104 made to model neural networks that could be trained to automatically recognize visual patterns with
105 modest performance ([Rosenblatt 1962](#); [Kabrisky 1966](#); [Giebel 1971](#); [Fukushima 1975](#)). However,

106 the breakthrough came with the “neocognitron” (Fukushima 1980), which was a self-learning
107 neural network for pattern recognition that was robust to changes in position and shape distortion, a
108 problem that plagued earlier efforts, including the “cognitron” also proposed by Fukushima (1975).

109 A few notable efforts demonstrated the neural networks for handwritten digit recognition
110 (Fukushima 1988; Denker et al. 1988), but these required significant preprocessing and feature
111 extraction. LeCun et al. (1989) soon afterward introduced a multilayer neural network that mapped
112 a feature in each neuron (representing a “local receptive field”) via convolution. This network could
113 also be trained by backpropagation like other existing neural networks and featured pooling opera-
114 tions for better distortion and translation invariance. Further developments from this milestone
115 yielded the LeNet-5 convolutional neural network which attained accuracy levels that rendered it
116 commercially viable.

117 The big data revolution coupled with technological advancements that have made it possible to
118 capture and store high resolution images have raised challenges that continue to be surmounted with
119 successively high-performing architectures. Over the past decade, some of these efforts resulted in
120 significant breakthroughs in performance. AlexNet (Krizhevsky et al. 2012), with 5 convolutional
121 layers and 3 dense layers—one of the largest CNNs of its time, won the ILSVRC-2012¹ competition
122 with a top-5 error rate of 15.3% and served as a landmark in the Deep Learning subdomain. Zeiler
123 and Fergus (2014) then introduced ZFNet, besting the performance of AlexNet, and pioneered
124 visualization techniques that were foundational for model inference and interpretability. In the
125 same year, GoogLeNet, a 22-layer network, was proposed (Szegedy et al. 2014), featuring the
126 novel “Inception module,” which allowed for efficiency and accuracy in a very deep network.
127 Subsequent improvements have been proposed to the original inception framework (Szegedy et al.
128 2015; Szegedy et al. 2016). VGGNet (Simonyan and Zisserman 2015) also pushed the boundaries
129 of depth with up 19 layers, achieving state-of-the-art performance at ILSVRC-2014. Finally, ResNet
130 (He et al. 2015) addressed the accuracy degradation problem that arises with increasing depth in a
131 network by successively fitting smaller sets of layers to the residual and employing skip connections.

¹ImageNet Large Scale Visual Recognition Challenge; held annually from 2010 through 2017.

132 With these innovations, an unprecedented level of depth was achieved. Implementations with with
133 34, 50, 101 and 152 layers were demonstrated. ResNet-152 won first place in ILSVRC-2015.

134 Along with these developments in their architectures, CNNs have demonstrated viability for
135 applications ranging from image classification, object and text detection to document tracking,
136 labeling, speech, among several other related fields (Gu et al. 2018). In this study, we show that
137 a relatively simple CNN architecture coupled with state-of-the-art approaches for model training
138 and regularization is capable of efficiently and effectively predicting tree failure classes.

139 DATA AND METHODS

140 Image data description

141 The training dataset consisted of 505 images, each having an original size of 4032×3024
142 pixels. Images were captured over a single field season in Massachusetts, USA, between May and
143 September 2020 to limit any potential influence of changes in tree appearance due to seasonal leaf
144 senescence on image processing. ESRI ArcMaps was used to randomly distribute sampling sites
145 across the state. Field assessments of trees to classify likelihood of failure followed the “Level
146 1” methods outlined in the second edition of the International Society of Arboriculture’s (ISA)
147 Tree Risk Assessment Best Management Practices (Smiley et al. 2017) and ISA’s Utility Tree Risk
148 Assessment Best Management Practices (Goodfellow 2020). This method is commonly used to
149 assess trees in the United States. A Level 1 assessment was selected for this study because: (1)
150 individual risk assessments may be prohibitively expensive at higher orders, i.e. Level 2 or Level 3
151 (Smiley et al. 2017), given the hundreds of thousands of trees utilities must manage across territory
152 areas; (2) utility right-of-way (ROW) easements may not allow utility inspectors full access to trees
153 in practical application of higher order risk assessment procedure if the trees are beyond the edge
154 of the ROW (Goodfellow 2020); and (3) studies have shown reasonable efficacy of limited basic
155 visual assessment techniques in identifying more severe tree defects (Rooney et al. 2005; Koeser
156 et al. 2016) leading to greater likelihood of failure ratings. The four categories of likelihood of tree
157 failure, which are always considered in a stated time frame, are defined as follows (Smiley et al.
158 2017):

- *Improbable*: failure unlikely either during normal or extreme weather conditions;
- *Possible*: failure expected under extreme weather conditions; but unlikely during normal weather conditions;
- *Probable*: failure expected under normal weather conditions within a given time frame;
- *Imminent*: failure has started or is most likely to occur in the near future, even if there is no significant wind or increased load. This is a rare occurrence for a risk assessor to encounter, and may require immediate action to protect targets from impact.

In this study, only images of trees assigned to the likelihood of failure categories of *Improbable*, *Possible* and *Probable* were included in modeling. Images of *Imminent* trees were excluded due to their rarity. Typical examples are shown for each category in Figure 1. In the original set of training images, the class distribution is given in Table 1.

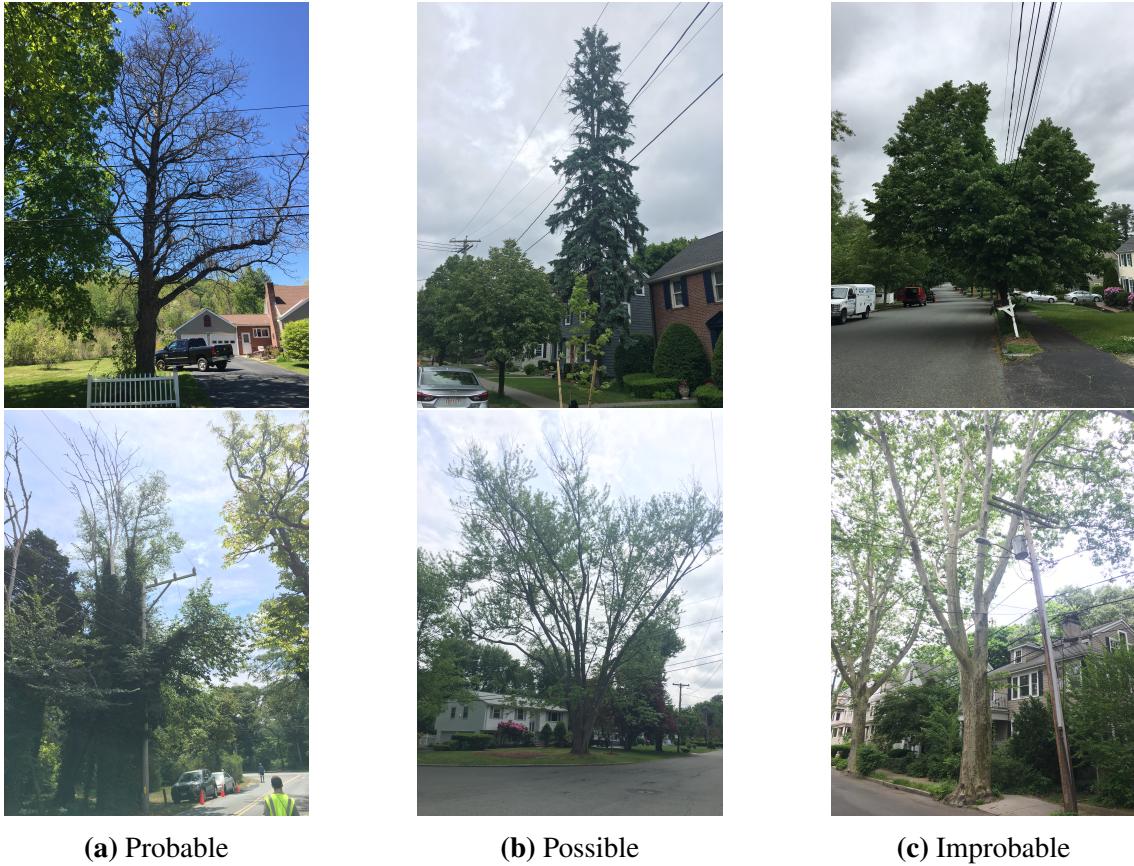
| Category | Number of images |
|-------------------|------------------|
| <i>Probable</i> | 56 |
| <i>Possible</i> | 80 |
| <i>Improbable</i> | 322 |
| Total | 505 |

TABLE 1. Category distribution in the set of raw input images

Classification scenarios

In order to investigate the efficacy of an AI classifier to distinguish the failure-liability categories, we defined four classification scenarios in Table 2 for our experiments. Each scenario represents a unique grouping of each of the three categories, with a minimum of two derived classes in each case.

Scenario Pr_Im considered only the highest and lowest likelihood of failure categories used in the study to clearly distinguish between categories. Since previous research has suggested that professionals more often disagree when distinguishing between possible and probable likelihood of failure ([Koeser et al. 2020](#)), in scenario PrPo_Im, we pooled trees in the *Probable* and *Possible* categories and compared them to trees in the *Improbable* category. In scenario Pr_PoIm, we



(a) Probable

(b) Possible

(c) Improbable

Fig. 1. Examples of training images in each of the three tree risk categories considered in this study. Trees in column (a) were categorized as *Improbable* due to their lack of structural defects as well as good physiological health. Trees in the center column were categorized as *Possible* due to weak branch unions and crown dieback. Trees in column (c) were categorized as *Probable* because they were dead. Leaves in the bottom image in the left-hand column are from vines attached to the dead tree.

| Scenario | Description | No. classes |
|----------|---|-------------|
| Pr_Im | { <i>Probable</i> , <i>Improbable</i> } | 2 |
| PrPo_Im | { <i>Probable</i> + <i>Possible</i> , <i>Improbable</i> } | 2 |
| Pr_PoIm | { <i>Probable</i> , <i>Possible</i> + <i>Improbable</i> } | 2 |
| Pr_Po_Im | { <i>Probable</i> , <i>Possible</i> , <i>Improbable</i> } | 3 |

TABLE 2. Classification scenarios

180 pooled trees in the *Possible* and *Improbable* categories and compared them to trees in the *Probable*
 181 category. In practice, this scenario is less likely because arborists typically distinguish trees with
 182 an *Improbable* likelihood of failure as those with minimal or no structural defects. It requires

183 additional judgment to distinguish trees with probable or possible likelihood of failure because
184 an arborist must assess the severity of structural defects, the presence of response growth, and
185 the expected loads (Smiley et al. 2017). Scenario Pr_Po_Im considered each likelihood of failure
186 category separately, as an arborist would do in practice.

187 **Image pre-processing and augmentation**

188 Data augmentation refers to the variety of methods that are employed for synthetically generating
189 more samples in a training dataset in order to improve model performance (Wong et al. 2016).
190 Augmentation is desired, particularly in situations where the number of original observations
191 is small, and the effectiveness of various relevant techniques in this domain has been amply
192 demonstrated (Shorten and Khoshgoftaar 2019).

193 In order to achieve robustness in our model, and given the relatively small number of training
194 images, we randomly cropped each image on either axis to 3024×3024 pixels, generating five
195 instances for each one. Thus, we increased the size of our training set from 505 to 2525 images.
196 Further, we performed horizontal flipping with a 50% probability on each of the generated images.
197 For efficiency, we converted the images to grayscale and scaled the pixel values from 0 to 1. Finally,
198 we downsampled the images to the following resolutions (pixels): 64×64 , 128×128 and 224×224 ,
199 creating a training set for each case. Random sets of images from each class across each of the four
200 classification scenarios are shown in Figure 2.

201 **Convolutional neural network**

202 We employ a convolutional neural network (CNN) as the AI framework for tree risk failure
203 likelihood prediction. Like other neural networks, the CNN is an arrangement of neurons within
204 layers, each neuron performing an operation that maps from a pixel in an input image to the final
205 output. The input into each neuron is a weighted sum from the previous layer, while the output from
206 each neuron is modulated by an activation function. The activation function in the final layer of an
207 CNN is typically the softmax function, which gives the class probabilities of a given input image.
208 A class is assigned to the image based on the one with the corresponding maximum probability.

209 Unlike other neural networks, however, the CNN performs fundamental pixel-mapping oper-

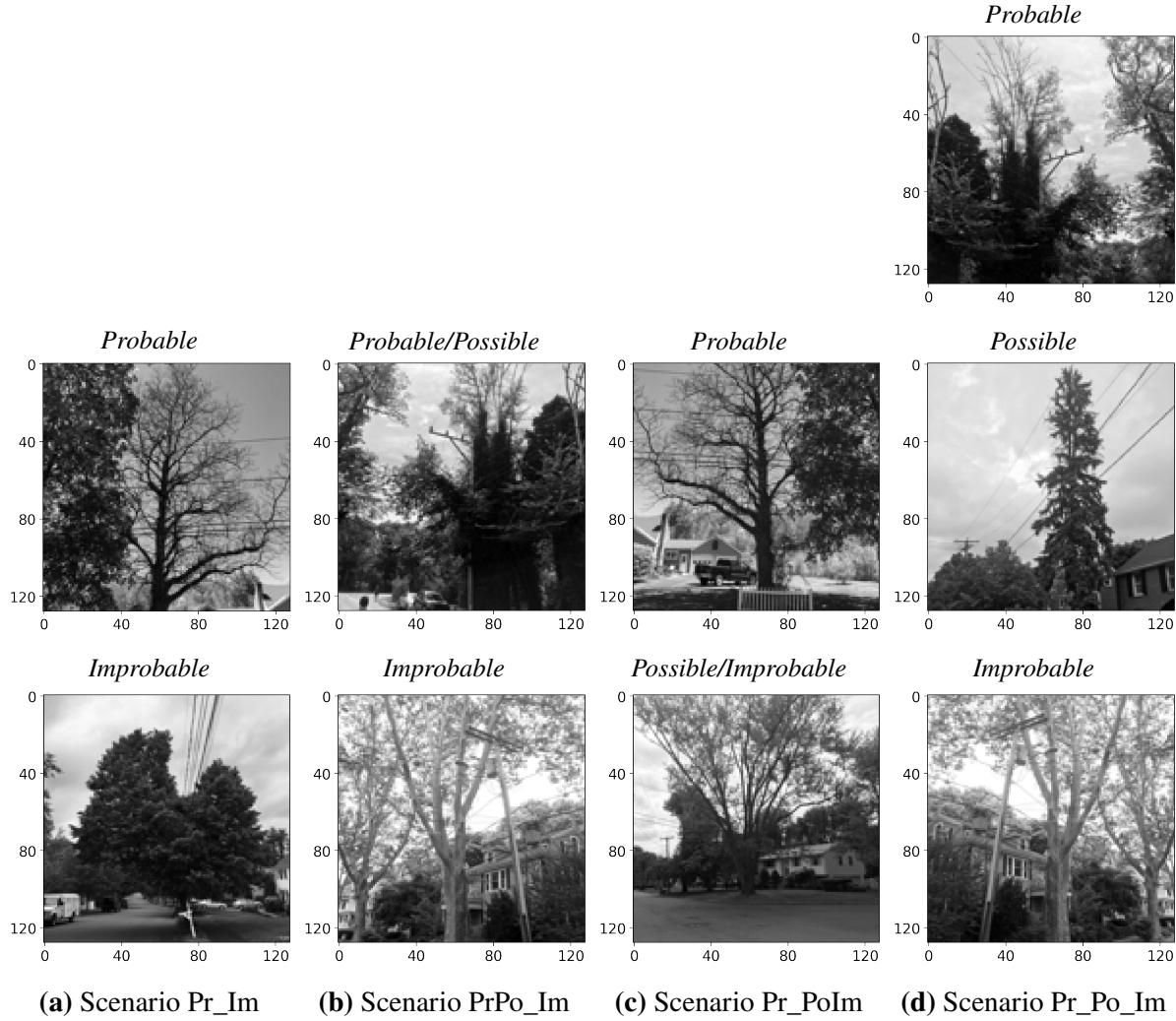


Fig. 2. Selected processed training images under each classification scenario. The images shown are processed versions of those shown in Figure 1. All were randomly cropped along the vertical axis and 50% were horizontally flipped (including some in this figure).

ations in its convolutional layers. Each convolutional layer is defined by a stack of feature maps, which result from a dot product of a filter and correspondingly-sized local receptive fields from the input image or preceding layer. The numeric values of the filters correspond to weights whose optimal values are learned during the training of the CNN. The size of the filter in each convolutional layer is given by the *kernel size*.

In this paper, we employ a relatively simple CNN architecture, historically inspired by AlexNet (Krizhevsky et al. 2012). The structure of the CNN is shown in Figure 3 (generated using an

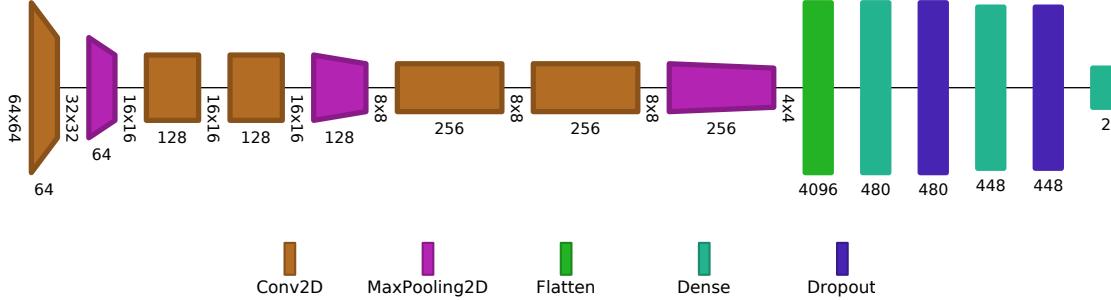


Fig. 3. Diagram of convolutional neural network structure (excluding the input layer). Hyperparameters that are optimized include the number of units in the penultimate dense layers. Here, 480 and 448 units are used, respectively.

automated framework (Bäuerle et al. 2021)). After the input layer (a matrix of pixels from the input image), we use a 64-filter convolutional layer. The output is downsampled using a pooling layer that returns the maximum output from a 2×2 subsample from the previous layer. Next, we stack two successive convolutional layers each with a depth of 128 filters. We follow these with a maximum-pooling layer and then two further 256-filter convolutional layers. A final maximum-pooling layer is used before we “flatten” all outputs into a one-dimensional (fully-connected) array of neurons. After flattening the outputs, we use a dense layer to reduce the number of outputs a specified number of units. Batch normalization (Ioffe and Szegedy 2015) is employed after the first dense layer to improve training efficiency. A second dense layer is used prior to the output layer, with the number of outputs corresponding to the number of classes in the dataset. A regularization technique known as “dropout” is used after each dense layer. In each dropout layer, a proportion of the neurons are randomly zeroed during training in order to improve the robustness of the model.

The various hyperparameters in the model are summarized in Table 3.

Hyperparameter optimization

We optimized eight of the CNN hyperparameters using Hyperband (Li et al. 2018), an efficient guided grid-search algorithm. Twelve searches were performed for each classification scenario and image resolution combination. Each search was conducted using 90 trials of unique hyperparameter combinations. The specified range of each parameter along with the search results are shown in Table 4. For the kernel size in the first convolutional layer, we allowed for a choice between a

| Layer | Layer # | No. Filters | Kernel Size | Strides | Activation | Rate | No. Units |
|---------------|---------|-------------|-------------|---------|------------|-------|-----------|
| Convolutional | 1 | 64 | k | 2 | ReLU | | |
| Max. Pooling | 1 | | 2 | | | | |
| Convolutional | 2 | 128 | 3 | 1 | ReLU | | |
| Convolutional | 3 | 128 | 3 | 1 | ReLU | | |
| Max. Pooling | 2 | | 2 | | | | |
| Convolutional | 4 | 256 | 3 | 1 | ReLU | | |
| Convolutional | 5 | 256 | 3 | 1 | ReLU | | |
| Max. Pooling | 3 | | 2 | | | | |
| Flatten | | | | | | | |
| Dense | 1 | | | | a_1 | | u_1 |
| Dropout | 1 | | | | | r_1 | |
| Dense | 2 | | | | a_2 | | u_2 |
| Dropout | 2 | | | | | r_2 | |

TABLE 3. Summary of the convolutional neural network hyperparameters. Those represented by a symbol are optimized using a guided search.

5 × 5 and a 7 × 7 kernel. The activation function in both dense layers was specified as a choice between the rectified linear unit (ReLU) function and the hyperbolic tangent (tanh). The ReLU was introduced to address the so-called “vanishing gradient” problem and has been shown to improve performance in CNNs (Glorot et al. 2011). Nevertheless, the tanh function remains a viable option, as well. The dropout rates were allowed to range from 0 to 5 in steps of 0.05, while the number of neurons or units in each dense layer varied from 32 to 512 in steps of 32. Finally, we uniformly sampled learning rates for the optimizer in the \log_{10} space of $[10^{-4}, 10^{-2}]$.

Model training and assessment

In this subsection, we provide an overview of the learning procedure for the convolutional neural network. As a reference, all the symbols used here are summarized in Table 5

The softmax activation function $f(\cdot)$ in the output layer returns the class prediction probabilities for a given observation i . It is defined in terms of the class-specific score s_c as:

$$f(s_c) = \frac{e^{s_c}}{\sum_{c' \in C} e^{s_{c'}}} \quad (1)$$

| Scenario | Hyperparameter | Range | Resolution | | |
|----------|------------------------------|------------------------|----------------------|----------------------|----------------------|
| | | | 64 | 128 | 224 |
| Pr_Im | 1st conv. kernel size, k | {5, 7} | 7 | 7 | 7 |
| | 1st dense activation, a_1 | {ReLU, tanh} | ReLU | ReLU | ReLU |
| | 2nd dense activation, a_2 | {ReLU, tanh} | ReLU | ReLU | ReLU |
| | 1st dropout rate, r_1 | {0, .05, ..., 5} | 0.1 | 0.1 | 0.1 |
| | 2nd dropout rate, r_2 | {0, .05, ..., 5} | 0.3 | 0.3 | 0.3 |
| | 1st dense layer units, u_1 | {32, 64, ..., 512} | 480 | 480 | 480 |
| | 2nd dense layer units, u_2 | {32, 64, ..., 512} | 448 | 448 | 448 |
| | learning rate, λ | [$10^{-4}, 10^{-2}$] | $1.03 \cdot 10^{-4}$ | $1.03 \cdot 10^{-4}$ | $1.03 \cdot 10^{-4}$ |
| PrPo_Im | 1st conv. kernel size, k | {5, 7} | 7 | 5 | 7 |
| | 1st dense activation, a_1 | {ReLU, tanh} | ReLU | tanh | ReLU |
| | 2nd dense activation, a_2 | {ReLU, tanh} | ReLU | ReLU | ReLU |
| | 1st dropout rate, r_1 | {0, .05, ..., 5} | 0.1 | 0.25 | 0.1 |
| | 2nd dropout rate, r_2 | {0, .05, ..., 5} | 0.3 | 0.35 | 0.3 |
| | 1st dense layer units, u_1 | {32, 64, ..., 512} | 480 | 384 | 480 |
| | 2nd dense layer units, u_2 | {32, 64, ..., 512} | 448 | 256 | 448 |
| | learning rate, λ | [$10^{-4}, 10^{-2}$] | $1.03 \cdot 10^{-4}$ | $1.09 \cdot 10^{-4}$ | $1.03 \cdot 10^{-4}$ |
| Pr_PoIm | 1st conv. kernel size, k | {5, 7} | 7 | 7 | 7 |
| | 1st dense activation, a_1 | {ReLU, tanh} | ReLU | ReLU | ReLU |
| | 2nd dense activation, a_2 | {ReLU, tanh} | tanh | ReLU | tanh |
| | 1st dropout rate, r_1 | {0, .05, ..., 5} | 0.25 | 0.1 | 0.2 |
| | 2nd dropout rate, r_2 | {0, .05, ..., 5} | 0.3 | 0.3 | 0.15 |
| | 1st dense layer units, u_1 | {32, 64, ..., 512} | 128 | 480 | 416 |
| | 2nd dense layer units, u_2 | {32, 64, ..., 512} | 320 | 448 | 416 |
| | learning rate, λ | [$10^{-4}, 10^{-2}$] | $1.76 \cdot 10^{-4}$ | $1.03 \cdot 10^{-4}$ | $1.15 \cdot 10^{-4}$ |
| Pr_Po_Im | 1st conv. kernel size, k | {5, 7} | 7 | 5 | 7 |
| | 1st dense activation, a_1 | {ReLU, tanh} | ReLU | tanh | ReLU |
| | 2nd dense activation, a_2 | {ReLU, tanh} | tanh | ReLU | ReLU |
| | 1st dropout rate, r_1 | {0, .05, ..., 5} | 0.25 | 0.25 | 0.1 |
| | 2nd dropout rate, r_2 | {0, .05, ..., 5} | 0.3 | 0.35 | 0.3 |
| | 1st dense layer units, u_1 | {32, 64, ..., 512} | 128 | 384 | 480 |
| | 2nd dense layer units, u_2 | {32, 64, ..., 512} | 320 | 256 | 448 |
| | learning rate, λ | [$10^{-4}, 10^{-2}$] | $1.76 \cdot 10^{-4}$ | $1.09 \cdot 10^{-4}$ | $1.03 \cdot 10^{-4}$ |

TABLE 4. Optimal hyperparameters found using the Hyperband search algorithm for 12 classification scenario and input image resolution combinations.

where C is the set of classes and c, c' are indices for a given class. Thus for the i th observation, the softmax activation returns the predicted probability $\hat{p}_{i,c}$ that the i th observation belongs to class c . The CNN is trained using a variant of the stochastic gradient algorithm, Adam (Kingma and Ba 2017). The goal of the training procedure is to learn the optimal weights and bias terms for

| Symbol | Definition |
|-----------------|---|
| c | Index of given class |
| $f(s_c)$ | Softmax activation function |
| F_{1c} | Class-specific F_1 score |
| F_1^m | Macro-average F_1 score |
| i | Index of given image observation |
| L_i^{CE} | Categorical cross entropy loss function of a single observation |
| $\hat{p}_{i,c}$ | Predicted probability that the i^{th} observation belongs to class c |
| Pr_c | Class-specific precision |
| Pr^m | Macro-average precision |
| Re_c | Class-specific recall |
| Re^m | Macro-average recall |
| s_c | Class-specific score |
| y_i | Observed (true) class of a given observation |
| \hat{y}_i | Predicted class of a given observation |

TABLE 5. Summary of symbols related to the training and assessment of the convolutional neural network

253 the CNN by minimizing a loss function. In this case, we use the categorical cross-entropy loss
 254 function, which for a single observation can be simply defined as:

255
$$L_i^{CE} = -\log(\hat{p}_{i,c}) = -\log(f(s_c^i)) \quad (2)$$

256 Training is iteratively performed, with gradient of the loss function computed and averaged over
 257 a batch of input images. Here, we use a batch size of 32. The learning rate of the optimization
 258 algorithm is an important hyperparameter that affects training performance. We optimized for this
 259 in the hyperparameter search as discussed. Furthermore, the CNN is trained over multiple passes
 260 through the entire training set. Each such pass is referred to as an epoch.

In real terms, we measured the performance of the trained CNN by how accurately it predicts the classes in a validation set excluded from the training set. For this paper, we used a randomly sampled validation that was 20% of the size of the input dataset of 2525 images in each training instance. Thus, we define the accuracy as the overall proportion of correct predictions across all classes. This metric was computed both for the training and validation sets in each epoch.

In addition to overall accuracy of making correct classifications, we assessed the models trained under these scenarios based on the macro-averages of the precision, recall and F_1 score metrics computed over the validation set in each epoch. The precision score Pr_c captures the proportion of correct predictions for a certain class relative to all the predictions for that class, and is an important measure of how good a classifier is. The recall Re_c captures the ability of a classifier to correctly predict observations for a certain class relative to all the true observations in that class. The F_{1c} metric is given as the class-specific harmonic mean of the precision and recall, and is thus more sensitive than the overall accuracy score. These three metrics are macro-averaged. Thus, each category is given equal weight, ensuring that misclassifications within the smaller classes (*Probable* and *Possible*) are adequately represented in the aggregation. These metrics are formally defined as follows:

$$\text{Macro-average precision: } Pr^m = \frac{1}{|C|} \sum_{c \in C} \left(\frac{TP_c}{TP_c + FP_c} \right) \quad (3)$$

$$\text{Macro-average recall: } Re^m = \frac{1}{|C|} \sum_{c \in C} \left(\frac{TP_c}{TP_c + FN_c} \right) \quad (4)$$

$$\text{Macro-average } F_1 \text{ score: } F_1^m = \frac{1}{|C|} \sum_{c \in C} \left(\frac{2Pr_c Re_c}{Pr_c + Re_c} \right) \quad (5)$$

where c is the index of a class in the set C and $|C|$ the number of classes in the dataset. The class-specific prediction metrics are given by:

$$\text{True positives for class } c: TP_c = \sum_{i \in c} I(\hat{y}_i = y_i) \quad (6)$$

$$\text{False positives for class } c: FP_c = \sum_{i \in c} I(\hat{y}_i \in c | y_i \notin c) \quad (7)$$

$$\text{False negatives for class } c: FN_c = \sum_{i \in c} I(\hat{y}_i \notin c | y_i \in c) \quad (8)$$

²⁶¹ where \hat{y}_i is the predicted class and y_i the observed (true) class for a given image i in class c . The
²⁶² indicator function $I(\cdot)$ returns 1 if the corresponding condition is true, and 0 otherwise.

263

RESULTS

264 We trained the CNN for each of the 4 classification scenarios. In each scenario, we also trained
 265 on three image resolution sets (64, 108 and 224). The goal was to determine the best performing
 266 resolution, given the tradeoff between performance and computational expenditure. Thus, we
 267 generated twelve learning cases. In each case, we performed a 5-fold cross-validation (resulting in
 268 a training-validation ratio of 80:20 in each fold). We trained the CNN over 20 epochs in all the
 269 cases. The trajectories of the loss and accuracy are shown in Figure 4 for both the training and
 270 validation sets.

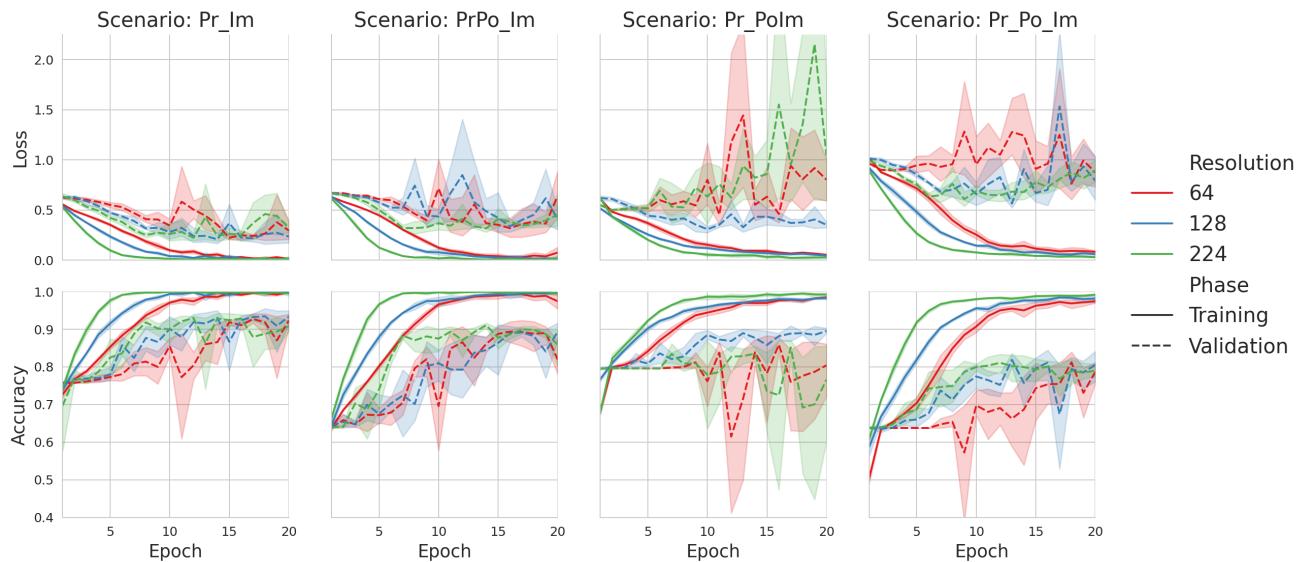


Fig. 4. Performance metrics for each scenario and input resolution instance. Average metrics and 95% confidence intervals (trials, $n = 5$) are shown for 20 epochs in each case.

271

Sensitivity to training resolution

272 Figure 5 shows the boxplots of the validation performance metrics for all twelve cases consid-
 273 ered. The metrics are: accuracy and the macro-averages of precision, recall and the F_1 score. For
 274 reference, we summarize the statistics (mean and standard deviation) of the validation performance
 275 metrics in Table 6. Taking all metrics into consideration, Welch's pairwise tests show that there are
 276 no significant differences in performance among the 3 training resolutions, with one exception. In
 277 the scenario Pr_PoIm, the accuracy, F_1^m , and Re^m for the 128-pixel case are greater than those for

278 the 224-pixel case ($p < .001$). This difference in performance is not as stark between the 128-pixel
279 and 64-pixel cases in the same scenario. This outcome implies that we can achieve efficiency by
280 training at lower resolutions without significant losses in performance. (A complete summary of
281 the p-values is provided in Table 7.)

282 Scenario sensitivity

283 We conducted pairwise tests between each scenario combination for all resolutions. (A complete
284 summary of the results is given in Table 8. The results indicated that considering all metrics,
285 the scenarios are statistically significantly different from each other ($p < .005$) except Pr_Im
286 compared to PrPo_Im and Pr_PoIm compared to Pr_Po_Im, with respect to certain metrics. If
287 the accuracy is not taken into account, then there is no strong evidence that Pr_Im differs in
288 performance when compared with PrPo_Im. In the case of Pr_PoIm versus Pr_Po_Im, there is
289 greater evidence to support their similarity in performance with respect to F_1^m and Re^m . Otherwise,
290 the performance between these two scenarios is significantly different. From Figure 5, we observe
291 the best performances in Pr_Im and PrPo_Im, with all metrics greater than or equal to 0.90. The
292 performance in Pr_PoIm is lower, and Pr_Po_Im has the lowest performance of all four.

293 Scenario Pr_Im unsurprisingly has the best performance (greatest accuracy and Pr^m of 0.95),
294 since the classifier only has to predict two extreme categories of *Probable* and *Improbable*. However,
295 the *Possible* category cannot be evaded in reality. Thus, from the perspective of practical application,
296 scenario PrPo_Im, which is the next best performing scenario (best accuracy of 0.92, $\hat{\sigma} = 0.01$),
297 demonstrates that it is most viable to group *Probable* with *Possible* for the best CNN performance.

298 Analysis of classification strategies

299 Given that there was no significant difference in performance among the three resolutions
300 tested, we focused on the 128-px case, as a tradeoff between efficiency and performance. We re-
301 trained the model on all four scenarios using this resolution and then evaluated performance on the
302 aforementioned validation metrics: accuracy and the macro-averages of class-specific precision,
303 recall and F_1 score. The goal of this analysis was to explore the efficiencies of various classification
304 strategies for tree failure risk.

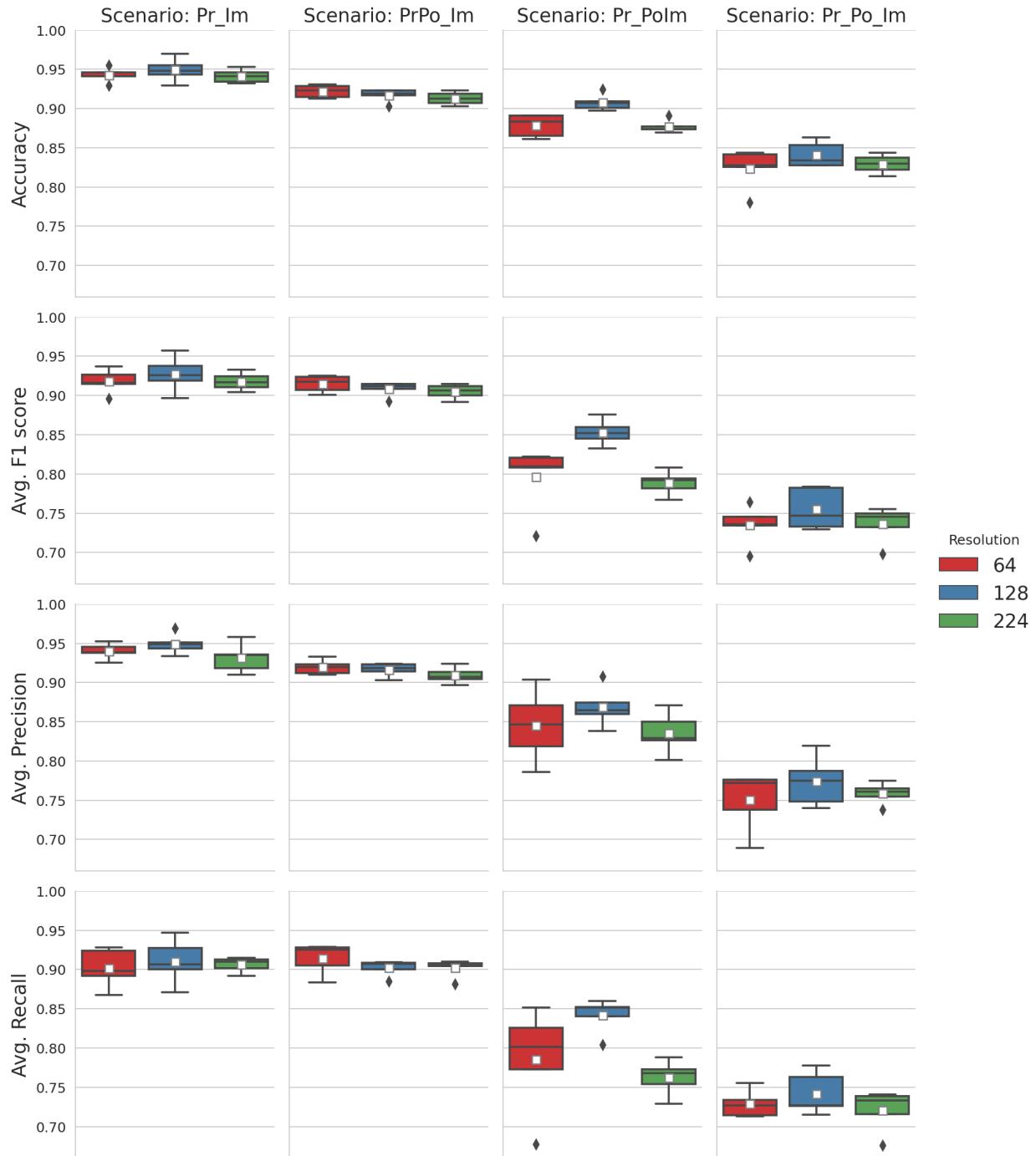


Fig. 5. Boxplots of validation performance metrics for each scenario and resolution combination. The upper and lower bounds of each box are the first and third quartiles; the whiskers are three standard deviations apart; and the diamonds are outliers. Mean values are depicted as white squares in each box.

Using a randomly sampled validation set that was 20% of the augmented dataset as before,
 we trained the CNN with the respective optimal hyperparameters for a single instance in order
 to compare the performance across the scenarios. The metrics are shown in Figure 6. In each
 scenario, we trained the model over 13 epochs. From this figure, we see that Pr_Po_Im performed
 significantly worse than the other three. This indicates the uncertainty surrounding the expert, yet
 subjective, failure likelihood assessment of the trees to begin with, particularly when the category
 is deemed to be *Possible*.

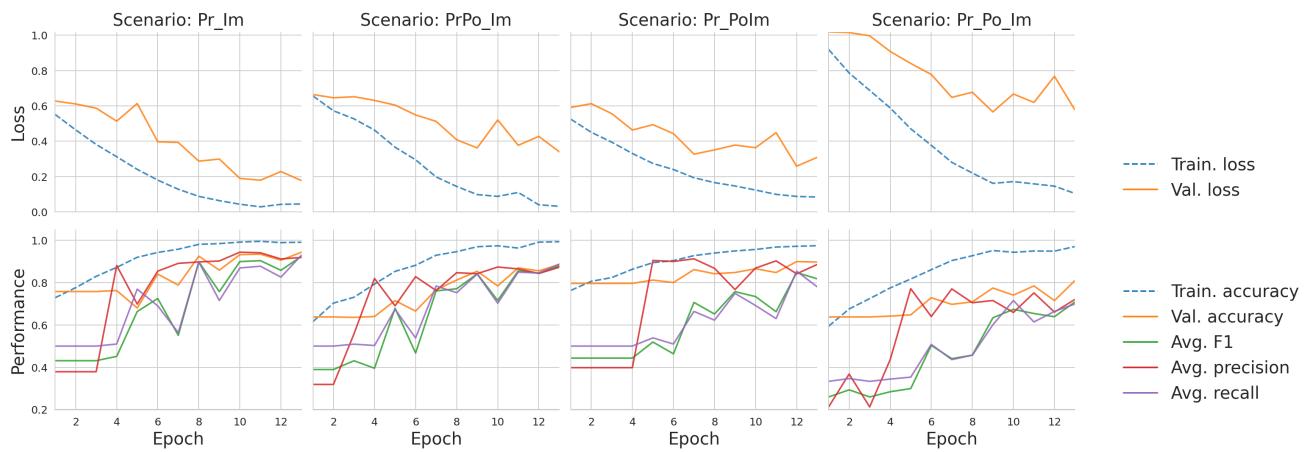


Fig. 6. Model performance across four classification scenarios for a single training instance in each case. Training resolution was 128px. All weighted average metrics were computed on the validation set.

We also plotted the confusion matrix for each scenario in Figure 7 to further investigate the
 performance of the classifier. Each row of the confusion matrix indicates the proportion of
 observations in a given class that are predicted to be in the classes across the columns. Thus, the
 diagonal entry in each matrix represents the class-specific recall score, Re_c . The matrices show that
 generally, the *Possible* category is difficult to predict accurately as an individual class. The classifier
 performs best when it only has to distinguish between *Probable* and *Improbable* (Figure 7a). This
 observation aligns with previous work (Koeser et al. 2020) and is arboriculturally intuitive. The
Improbable category includes trees with no apparent structural defects, which is comparatively
 easy for a qualified arborist to assess. When a defect is present, however, an arborist must estimate
 the severity of the defect that is present, whether the tree has grown in such a way that structurally

compensates for the presence of the defect, and the anticipated loads on the tree. Although there are guidelines to inform an arborist’s assessment (Smiley et al. 2017; Goodfellow 2020), evaluating each of these factors still depends on an arborist’s judgment and experience, making it more difficult to distinguish between trees in the *Possible* and *Probable* categories. Thus, it was not surprising that performance only suffered slightly when *Possible* was grouped with *Probable* in the PrPo_Im scenario (Figure 7b), which we would select as the best strategy from a practical standpoint. These two scenarios (Pr_Im and PrPo_Im) have the highest class-specific recall scores Re_c (≥ 0.90) compared to the other two scenarios Pr_PoIm ($Re_c \geq 0.58$) and Pr_Po_Im ($Re_c \geq 0.54$). In scenario Pr_PoIm, we see that even grouping *Possible* with *Improbable* worsens the ability of the classifier to distinguish the *Probable* category.

CONCLUSION

We have demonstrated the efficacy of an artificial intelligence framework for predicting tree failure likelihood (*Probable*, *Possible* and *Improbable*) with respect to utility infrastructure. Specifically, we developed a convolutional neural network with state-of-the-art configurations. We applied data augmentation and pre-processing strategies to increase the size of our dataset by a factor of 5, thus generating 2525 images. From an initial resolution of 4032×3024 pixels, we created three sets of image resolutions: 64×64 , 128×128 and 224×224 . We then defined four classification scenarios to investigate the performance of various groupings of the three categories: Pr_Im (*Probable* vs. *Improbable*); PrPo_Im (*Probable* and *Possible* vs. *Improbable*); Pr_PoIm (*Probable* vs. *Possible* and *Improbable*); and Pr_Po_Im (*Probable* vs. *Possible* vs. *Improbable*). We then optimized eight of our CNN hyperparameters for 12 classification scenario and image resolution combinations. We conducted five-fold cross-validation for each of the 12 cases, and assessed model performance based on accuracy and the macro-averages of precision, recall and F_1 score.

Our results indicated that there generally was no significant difference in performance between the resolutions. Among the scenarios, however, Pr_Im performed the best with a top accuracy and Pr^m of 0.94 ($\hat{\sigma} = 0.01$). The next best-performing scenario was PrPo_Im with a top accuracy and Pr^m of 0.92 ($\hat{\sigma} = 0.01$). For practical applications, scenario PrPo_Im is more realistic, as it

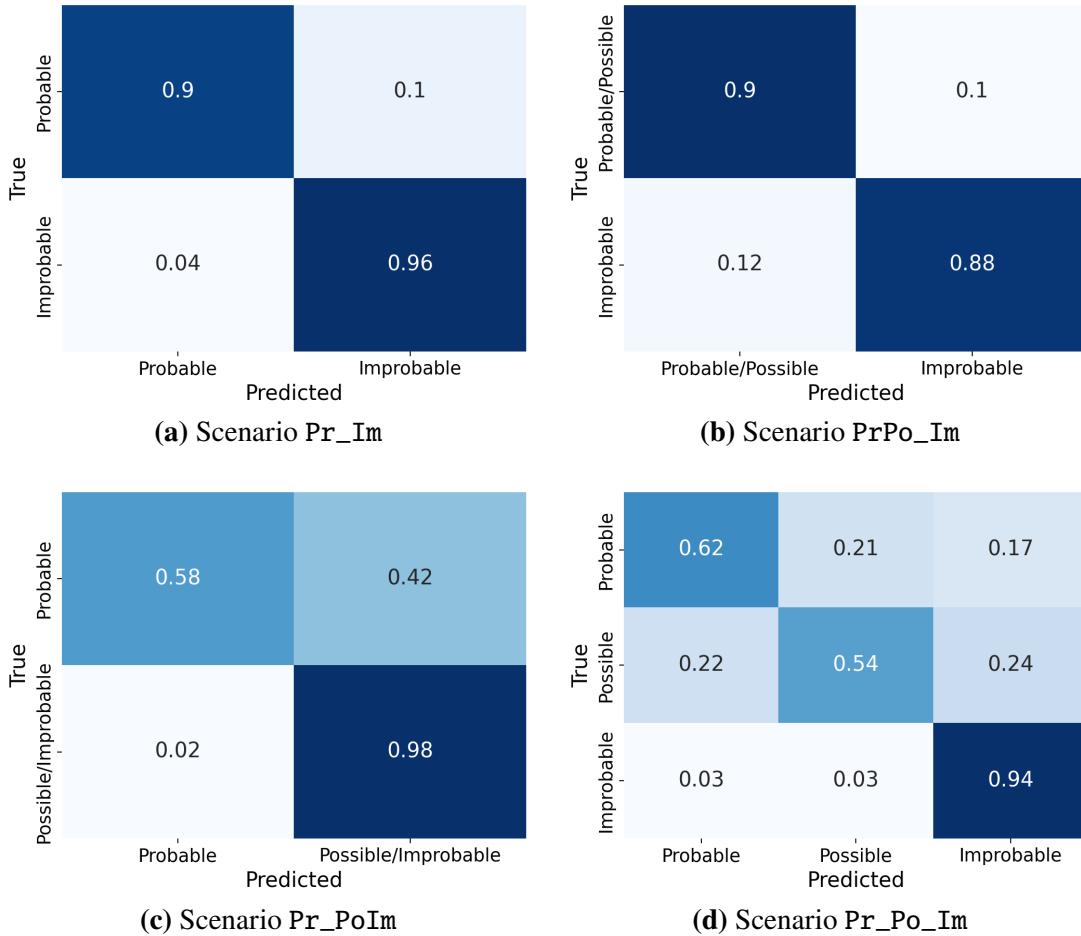


Fig. 7. Scenario confusion matrices for a single training instance using the 128-pixel images. Each row of a confusion matrix indicates the proportional distribution of class predictions for the true members of each class. Thus, the diagonals indicate the recall for each class Re_c . The average of the diagonal values gives the macro-average Re^m for each scenario.

349 includes all three categories. Thus, we deemed this as the most viable. Scenarios Pr_PoIm and
 350 Pr_Po_Im had best accuracy scores of 0.91 ($\hat{\sigma} = 0.01$) and 0.84 ($\hat{\sigma} = 0.02$), respectively. But their
 351 performance across the other metrics was considerably worse.

352 We further conducted more detailed classification performance analyses over single training
 353 instances for the 128-pixel case. Our results indicate that the CNN performed best at recalling
 354 *Probable* vs. *Improbable* or *Probable/Possible* vs. *Improbable*. The *Possible* category appeared to
 355 be a confounding for the classifier. This was not unexpected, given the uncertainty and subjectivity
 356 in assessing these trees to begin with. The CNN, however, performed better at distinguishing

357 between *Possible/Improbable* and *Probable*, than between *Probable/Possible* and *Improbable*.
358 This might be an indicator that trees in the *Possible* category are more likely to be identified as
359 *Improbable*. When all three failure-liability categories were predicted individually in scenario
360 Pr_Po_Im, the *Possible* category had the lowest class-specific recall score.

361 Nevertheless, given the relatively small input dataset of original images, these preliminary
362 results are extremely promising for future improvements. First, we can train better models with
363 more data. We also plan to rigorously quantify the uncertainty in ground truth category assignments
364 by incorporating predictions from multiple experts on the same images. In order to better understand
365 how the CNN is classifying each image, we will conduct extensive visual inference in further work,
366 for instance, using the gradient-weighted class activation mapping approach ([Zeiler and Fergus
367 2014](#)). There is also a potential for mapping the visual cues learned by the CNN to physical
368 relationships governing tree structure, in order to gain greater insights into tree failure processes.
369 The automation for tree failure likelihood assessments can potentially supplement human decision-
370 making for increased resilience and, in the future, reduce costs and improve reliability in tree
371 assessments, thus leading to more sustainable communities.

372 DATA AVAILABILITY STATEMENT

373 All data, models, or code generated or used during the study are publicly available in a GitHub
374 repository online at <https://github.com/narslab/tree-risk-ai>.

375 ACKNOWLEDGMENTS

376 The authors acknowledge the partial support of Eversource Energy in funding this work.

APPENDIX I. SUMMARY OF VALIDATION PERFORMANCE METRICS

378 For further reference we tabulate the mean and standard deviation of the performance metrics
 379 obtained on the scenario-resolution sensitivity tests. These values are shown in Table 6. The
 380 metrics are: accuracy, macro-average F_1 (F_1^m), macro-average recall (Re^m) and macro-average
 381 precision (Pr^m).

TABLE 6. Mean and standard deviation of the 5-fold cross-validation estimates of classifier performance metrics on the 12 scenario-resolution cases investigated. “SD” represents “standard deviation.”

| Scenario | Resolution | Accuracy | | F_1^m | | Re^m | | Pr^m | |
|-----------------|-------------------|-----------------|------|---------------------------|------|--------------------------|------|--------------------------|------|
| | | Mean | SD | Mean | SD | Mean | SD | Mean | SD |
| Pr_Im | 64 | 0.94 | 0.01 | 0.92 | 0.02 | 0.9 | 0.02 | 0.94 | 0.01 |
| | 128 | 0.95 | 0.01 | 0.93 | 0.02 | 0.91 | 0.03 | 0.95 | 0.01 |
| | 224 | 0.94 | 0.01 | 0.92 | 0.01 | 0.91 | 0.01 | 0.93 | 0.02 |
| PrPo_Im | 64 | 0.92 | 0.01 | 0.91 | 0.01 | 0.91 | 0.02 | 0.92 | 0.01 |
| | 128 | 0.92 | 0.01 | 0.91 | 0.01 | 0.9 | 0.01 | 0.92 | 0.01 |
| | 224 | 0.91 | 0.01 | 0.91 | 0.01 | 0.9 | 0.01 | 0.91 | 0.01 |
| Pr_PoIm | 64 | 0.88 | 0.01 | 0.8 | 0.04 | 0.79 | 0.07 | 0.85 | 0.05 |
| | 128 | 0.91 | 0.01 | 0.85 | 0.02 | 0.84 | 0.02 | 0.87 | 0.03 |
| | 224 | 0.88 | 0.01 | 0.79 | 0.02 | 0.76 | 0.02 | 0.84 | 0.03 |
| Pr_Po_Im | 64 | 0.82 | 0.03 | 0.74 | 0.03 | 0.73 | 0.02 | 0.75 | 0.04 |
| | 128 | 0.84 | 0.02 | 0.76 | 0.03 | 0.74 | 0.03 | 0.77 | 0.03 |
| | 224 | 0.83 | 0.01 | 0.74 | 0.02 | 0.72 | 0.03 | 0.76 | 0.01 |

382 **APPENDIX II. WELCH T -TEST RESULTS**

383 Here, we summarize the p-values resulting from the Welch independent t -tests used to determine
 384 if the mean values of the various performance metrics are statistically significant or not. Strong
 385 evidence of significant difference is given by a low p-value (a typical threshold is < 0.01). In Table 7,
 386 we consider pairwise comparisons between different image resolutions in all four scenarios. In
 387 Table 8, we compare different scenarios for the same image resolution.

TABLE 7. Welch's t -test p-value results for pairwise comparisons between image resolutions across all four scenarios. Low p-values indicates statistically different mean values for the validation metrics considered.

| Scenario | Resolution 1 | Resolution 2 | p-values | | | |
|----------|--------------|--------------|----------------------|----------------------|----------------------|----------------------|
| | | | Accuracy | F_1^m | Re^m | Pr^m |
| Pr_Im | 64 | 128 | $4.28 \cdot 10^{-1}$ | $4.72 \cdot 10^{-1}$ | $6.38 \cdot 10^{-1}$ | $2.43 \cdot 10^{-1}$ |
| | 64 | 224 | $8.11 \cdot 10^{-1}$ | $9.58 \cdot 10^{-1}$ | $7.29 \cdot 10^{-1}$ | $4.01 \cdot 10^{-1}$ |
| | 128 | 224 | $3.33 \cdot 10^{-1}$ | $4.21 \cdot 10^{-1}$ | $7.82 \cdot 10^{-1}$ | $1.19 \cdot 10^{-1}$ |
| PrPo_Im | 64 | 128 | $3.43 \cdot 10^{-1}$ | $3.35 \cdot 10^{-1}$ | $2.80 \cdot 10^{-1}$ | $6.07 \cdot 10^{-1}$ |
| | 64 | 224 | $1.13 \cdot 10^{-1}$ | $1.53 \cdot 10^{-1}$ | $3.00 \cdot 10^{-1}$ | $1.32 \cdot 10^{-1}$ |
| | 128 | 224 | $4.65 \cdot 10^{-1}$ | $5.67 \cdot 10^{-1}$ | $9.78 \cdot 10^{-1}$ | $2.54 \cdot 10^{-1}$ |
| Pr_PoIm | 64 | 128 | $6.93 \cdot 10^{-3}$ | $3.78 \cdot 10^{-2}$ | $1.39 \cdot 10^{-1}$ | $3.45 \cdot 10^{-1}$ |
| | 64 | 224 | $8.37 \cdot 10^{-1}$ | $7.16 \cdot 10^{-1}$ | $4.95 \cdot 10^{-1}$ | $6.97 \cdot 10^{-1}$ |
| | 128 | 224 | $1.10 \cdot 10^{-3}$ | $2.07 \cdot 10^{-4}$ | $4.67 \cdot 10^{-4}$ | $7.62 \cdot 10^{-2}$ |
| Pr_Po_Im | 64 | 128 | $2.42 \cdot 10^{-1}$ | $2.50 \cdot 10^{-1}$ | $3.91 \cdot 10^{-1}$ | $3.12 \cdot 10^{-1}$ |
| | 64 | 224 | $6.77 \cdot 10^{-1}$ | $9.38 \cdot 10^{-1}$ | $5.98 \cdot 10^{-1}$ | $6.50 \cdot 10^{-1}$ |
| | 128 | 224 | $2.28 \cdot 10^{-1}$ | $2.58 \cdot 10^{-1}$ | $2.53 \cdot 10^{-1}$ | $3.67 \cdot 10^{-1}$ |

TABLE 8. Welch's t -test p-value results for pairwise comparisons between scenarios resolutions for all three image resolutions. Low p-values indicates statistically different mean values for the validation metrics considered.

| Scenario | Resolution 1 | Resolution 2 | p-values | | | |
|----------|--------------|--------------|----------------------|----------------------|----------------------|----------------------|
| | | | Accuracy | F_1^m | Re^m | Pr^m |
| 64 | Pr_Im | PrPo_Im | $5.91 \cdot 10^{-3}$ | $7.14 \cdot 10^{-1}$ | $4.23 \cdot 10^{-1}$ | $1.06 \cdot 10^{-2}$ |
| | Pr_Im | Pr_PoIm | $6.80 \cdot 10^{-5}$ | $1.77 \cdot 10^{-3}$ | $1.48 \cdot 10^{-2}$ | $8.35 \cdot 10^{-3}$ |
| | Pr_Im | Pr_Po_Im | $1.84 \cdot 10^{-4}$ | $4 \cdot 10^{-6}$ | $4 \cdot 10^{-6}$ | $1.93 \cdot 10^{-4}$ |
| | PrPo_Im | Pr_PoIm | $8.14 \cdot 10^{-4}$ | $2.53 \cdot 10^{-3}$ | $1.08 \cdot 10^{-2}$ | $2.02 \cdot 10^{-2}$ |
| | PrPo_Im | Pr_Po_Im | $5.53 \cdot 10^{-4}$ | $1.40 \cdot 10^{-5}$ | 0. | $3.51 \cdot 10^{-4}$ |
| | Pr_PoIm | Pr_Po_Im | $5.38 \cdot 10^{-3}$ | $2.91 \cdot 10^{-2}$ | $1.32 \cdot 10^{-1}$ | $7.42 \cdot 10^{-3}$ |
| 128 | Pr_Im | PrPo_Im | $4.71 \cdot 10^{-3}$ | $1.38 \cdot 10^{-1}$ | $5.79 \cdot 10^{-1}$ | $2.39 \cdot 10^{-3}$ |
| | Pr_Im | Pr_PoIm | $1.25 \cdot 10^{-3}$ | $4.55 \cdot 10^{-4}$ | $3.30 \cdot 10^{-3}$ | $7.90 \cdot 10^{-4}$ |
| | Pr_Im | Pr_Po_Im | $5 \cdot 10^{-6}$ | $4 \cdot 10^{-6}$ | $1.20 \cdot 10^{-5}$ | $6.00 \cdot 10^{-5}$ |
| | PrPo_Im | Pr_PoIm | $1.70 \cdot 10^{-1}$ | $4.46 \cdot 10^{-4}$ | $1.73 \cdot 10^{-3}$ | $1.13 \cdot 10^{-2}$ |
| | PrPo_Im | Pr_Po_Im | $9.90 \cdot 10^{-5}$ | $6.40 \cdot 10^{-5}$ | $5.00 \cdot 10^{-5}$ | $3.20 \cdot 10^{-4}$ |
| | Pr_PoIm | Pr_Po_Im | $1.32 \cdot 10^{-4}$ | $2.44 \cdot 10^{-4}$ | $2.44 \cdot 10^{-4}$ | $9.31 \cdot 10^{-4}$ |
| 224 | Pr_Im | PrPo_Im | $7.16 \cdot 10^{-4}$ | $8.86 \cdot 10^{-2}$ | $5.77 \cdot 10^{-1}$ | $5.50 \cdot 10^{-2}$ |
| | Pr_Im | Pr_PoIm | $2 \cdot 10^{-6}$ | $1 \cdot 10^{-6}$ | $2.50 \cdot 10^{-5}$ | $2.56 \cdot 10^{-4}$ |
| | Pr_Im | Pr_Po_Im | 0. | $5 \cdot 10^{-6}$ | $2.90 \cdot 10^{-5}$ | 0. |
| | PrPo_Im | Pr_PoIm | $1.30 \cdot 10^{-4}$ | $3 \cdot 10^{-6}$ | $1.40 \cdot 10^{-5}$ | $1.83 \cdot 10^{-3}$ |
| | PrPo_Im | Pr_Po_Im | $3 \cdot 10^{-6}$ | $1.40 \cdot 10^{-5}$ | $1.70 \cdot 10^{-5}$ | 0. |
| | Pr_PoIm | Pr_Po_Im | $1.44 \cdot 10^{-4}$ | $3.62 \cdot 10^{-3}$ | $2.93 \cdot 10^{-2}$ | $1.15 \cdot 10^{-3}$ |

REFERENCES

- Bäuerle, A., van Onzenoodt, C., and Ropinski, T. (2021). “Net2Vis – A Visual Grammar for Automatically Generating Publication-Tailored CNN Architecture Visualizations.” *IEEE Transactions on Visualization and Computer Graphics*, 1–1.
- Denker, J., Gardner, W., Graf, H., Henderson, D., Howard, R., Hubbard, W., Jackel, L. D., Baird, H., and Guyon, I. (1988). “Neural Network Recognizer for Hand-Written Zip Code Digits.” *Advances in Neural Information Processing Systems*, 1, 323–331.
- dos Santos, A. A., Marcato Junior, J., Araújo, M. S., Di Martini, D. R., Tetila, E. C., Siqueira, H. L., Aoki, C., Eltner, A., Matsubara, E. T., Pistori, H., Feitosa, R. Q., Liesenberg, V., and Gonçalves, W. N. (2019). “Assessment of CNN-Based Methods for Individual Tree Detection on Images Captured by RGB Cameras Attached to UAVs.” *Sensors*, 19(16), 3595.

- 399 Egli, S. and Höpke, M. (2020). “CNN-Based Tree Species Classification Using High Resolution
400 RGB Image Data from Automated UAV Observations.” *Remote Sensing*, 12(23), 3892.
- 401 Fricker, G. A., Ventura, J. D., Wolf, J. A., North, M. P., Davis, F. W., and Franklin, J. (2019). “A
402 Convolutional Neural Network Classifier Identifies Tree Species in Mixed-Conifer Forest from
403 Hyperspectral Imagery.” *Remote Sensing*, 11(19), 2326.
- 404 Fukushima, K. (1975). “Cognitron: A self-organizing multilayered neural network.” *Biological
405 Cybernetics*, 20(3), 121–136.
- 406 Fukushima, K. (1980). “Neocognitron: A self-organizing neural network model for a mechanism
407 of pattern recognition unaffected by shift in position.” *Biological Cybernetics*, 36(4), 193–202.
- 408 Fukushima, K. (1988). “Neocognitron: A hierarchical neural network capable of visual pattern
409 recognition.” *Neural Networks*, 1(2), 119–130.
- 410 Giebel, H. (1971). “Feature Extraction and Recognition of Handwritten Characters by Homoge-
411 neous Layers.” *Zeichenerkennung Durch Biologische Und Technische Systeme / Pattern Recog-
412 nition in Biological and Technical Systems*, O.-J. Grüsser and R. Klinke, eds., Berlin, Heidelberg,
413 Springer, 162–169.
- 414 Glorot, X., Bordes, A., and Bengio, Y. (2011). “Deep Sparse Rectifier Neural Networks.” *Proceed-
415 ings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, JMLR
416 Workshop and Conference Proceedings, 315–323 (June).
- 417 Goodfellow, J. W. (2020). “Best Management Practices - Utility Tree Risk Assessment.” *Report
418 No. P1321*, International Society of Arboriculture.
- 419 Graziano, M., Gunther, P., Gallaher, A., Carstensen, F. V., and Becker, B. (2020). “The wider
420 regional benefits of power grids improved resilience through tree-trimming operations evidences
421 from Connecticut, USA.” *Energy Policy*, 138, 111293.
- 422 Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J.,
423 and Chen, T. (2018). “Recent advances in convolutional neural networks.” *Pattern Recognition*,
424 77, 354–377.
- 425 Guggenmoos, S. (2003). “EFFECTS OF TREE MORTALITY ON POWER LINE SECURITY.”

- 426 *Journal of Arboriculture*, 29(4), 181–196.
- 427 Guggenmoos, S. (2011). “Tree-related Electric Outages Due To Wind Loading.” *Arboriculture and*
428 *Urban Forestry*, 37(4), 147–151.
- 429 He, K., Zhang, X., Ren, S., and Sun, J. (2015). “Deep Residual Learning for Image Recognition.”
430 *arXiv:1512.03385 [cs]*.
- 431 Hubel, D. H. and Wiesel, T. N. (1959). “Receptive fields of single neurones in the cat’s striate
432 cortex.” *The Journal of Physiology*, 148(3), 574–591.
- 433 Hubel, D. H. and Wiesel, T. N. (1962). “Receptive fields, binocular interaction and functional
434 architecture in the cat’s visual cortex.” *The Journal of Physiology*, 160(1), 106–154.2.
- 435 Hubel, D. H. and Wiesel, T. N. (1965). “Receptive fields and functional architecture in two nonstriate
436 visual areas (18 and 19) of the cat.” *Journal of Neurophysiology*, 28(2), 229–289.
- 437 Hubel, D. H. and Wiesel, T. N. (1968). “Receptive fields and functional architecture of monkey
438 striate cortex.” *The Journal of Physiology*, 195(1), 215–243.
- 439 Ioffe, S. and Szegedy, C. (2015). “Batch Normalization: Accelerating Deep Network Training by
440 Reducing Internal Covariate Shift.” *arXiv:1502.03167 [cs]*.
- 441 Jiao, P. and Alavi, A. H. (2020). “Artificial intelligence in seismology: Advent, performance and
442 future trends.” *Geoscience Frontiers*, 11(3), 739–744.
- 443 Kabrisky, M. (1966). *A Proposed Model for Visual Information Processing in the Human Brain*.
444 University of Illinois Press.
- 445 Kingma, D. P. and Ba, J. (2017). “Adam: A Method for Stochastic Optimization.” *arXiv:1412.6980*
446 *[cs]*.
- 447 Koeser, A. K., McLean, D. C., Hasing, G., and Allison, R. B. (2016). “Frequency, severity,
448 and detectability of internal trunk decay of street tree Quercus spp. in Tampa, Florida, U.S..”
449 *Arboriculture & Urban Forestry*, 42(4), 217–226.
- 450 Koeser, A. K., Thomas Smiley, E., Hauer, R. J., Kane, B., Klein, R. W., Landry, S. M., and
451 Sherwood, M. (2020). “Can professionals gauge likelihood of failure? – Insights from tropical
452 storm Matthew.” *Urban Forestry & Urban Greening*, 52, 126701.

- 453 Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). “ImageNet Classification with Deep
454 Convolutional Neural Networks.” *Advances in Neural Information Processing Systems*, 25, 1097–
455 1105.
- 456 LeCun, Y., Boser, B. E., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. E., and Jackel,
457 L. D. (1989). “Handwritten Digit Recognition with a Back-Propagation Network.” *Advances in
458 Neural Information Processing Systems* 2, 9.
- 459 Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., and Talwalkar, A. (2018). “Hyperband: A
460 Novel Bandit-Based Approach to Hyperparameter Optimization.” *Journal of Machine Learning
461 Research*, 18(185), 1–52.
- 462 Nateghi, R., Guikema, S., and Quiring, S. M. (2014). “Power Outage Estimation for Tropical
463 Cyclones: Improved Accuracy with Simpler Models.” *Risk Analysis*, 34(6), 1069–1078.
- 464 Parent, J. R., Meyer, T. H., Volin, J. C., Fahey, R. T., and Witharana, C. (2019). “An analysis of
465 enhanced tree trimming effectiveness on reducing power outages.” *Journal of Environmental
466 Management*, 241, 397–406.
- 467 Rooney, C. J., Ryan, H. D., Bloniarz, D. V., and Kane, B. (2005). “THE RELIABILITY OF
468 A WINDSHIELD SURVEY TO LOCATE HAZARDS IN ROADSIDE TREES.” *Journal of
469 Arboriculture*, 31(2).
- 470 Rosenblatt, F. (1962). *Principles of Neurodynamics; Perceptrons and the Theory of Brain Mechanisms*. Spartan Books, Washington.
- 471 Salehi, H. and Burgueño, R. (2018). “Emerging artificial intelligence methods in structural engi-
472 neering.” *Engineering Structures*, 171, 170–189.
- 473 Shorten, C. and Khoshgoftaar, T. M. (2019). “A survey on Image Data Augmentation for Deep
474 Learning.” *Journal of Big Data*, 6(1), 60.
- 475 Simonyan, K. and Zisserman, A. (2015). “Very Deep Convolutional Networks for Large-Scale
476 Image Recognition.” *arXiv:1409.1556 [cs]*.
- 477 Simpson, P. and Van Bossuyt, R. (1996). “TREE-CAUSED ELECTRIC OUTAGES.” *Journal of
478 Arboriculture*, 22, 117–121.

- 480 Smiley, E. T., Matheny, N., and Lilly, S. (2017). “Best Management Practices - Tree Risk Assess-
481 ment, Second Edition.” *Report No. P1542*, International Society of Arboriculture.
- 482 Spencer, B. F., Hoskere, V., and Narazaki, Y. (2019). “Advances in Computer Vision-Based Civil
483 Infrastructure Inspection and Monitoring.” *Engineering*, 5(2), 199–222.
- 484 Su, Z., Chow, J. K., Tan, P. S., Wu, J., Ho, Y. K., and Wang, Y.-H. (2020). “Deep convolutional
485 neural network-based pixel-wise landslide inventory mapping.” *Landslides*.
- 486 Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. (2016). “Inception-v4, Inception-ResNet and
487 the Impact of Residual Connections on Learning.” *arXiv:1602.07261 [cs]*.
- 488 Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and
489 Rabinovich, A. (2014). “Going Deeper with Convolutions.” *arXiv:1409.4842 [cs]*.
- 490 Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2015). “Rethinking the Inception
491 Architecture for Computer Vision.” *arXiv:1512.00567 [cs]*.
- 492 Wang, N., Zhao, X., Wang, L., and Zou, Z. (2019). “Novel System for Rapid Investigation and
493 Damage Detection in Cultural Heritage Conservation Based on Deep Learning.” *Journal of*
494 *Infrastructure Systems*, 25(3), 04019020.
- 495 Wismer, S. (2018). “Targeted Tree Trimming Offers Reliability Benefits.” *T&D World* (May).
- 496 Wong, S. C., Gatt, A., Stamatescu, V., and McDonnell, M. D. (2016). “Understanding data aug-
497 mentation for classification: When to warp?” *arXiv:1609.08764 [cs]*.
- 498 Xie, Y., Ebad Sichani, M., Padgett, J. E., and DesRoches, R. (2020). “The promise of implementing
499 machine learning in earthquake engineering: A state-of-the-art review.” *Earthquake Spectra*,
500 36(4), 1769–1801.
- 501 Zeiler, M. D. and Fergus, R. (2014). “Visualizing and Understanding Convolutional Networks.”
502 *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds., Lecture
503 Notes in Computer Science, Cham, Springer International Publishing, 818–833.