Taylor & Francis
Taylor & Francis Group

Sustainable and Resilient Infrastructure

ISSN: 2378-9689 (Print) 2378-9697 (Online) Journal homepage: www.tandfonline.com/journals/tsri20

Predicting tree failure likelihood for utility risk mitigation via a convolutional neural network

Atanas Apostolov, Jimi Oke, Ryan Suttle, Sanjay Arwade & Brian Kane

To cite this article: Atanas Apostolov, Jimi Oke, Ryan Suttle, Sanjay Arwade & Brian Kane (2023) Predicting tree failure likelihood for utility risk mitigation via a convolutional neural network, Sustainable and Resilient Infrastructure, 8:6, 572-588, DOI: 10.1080/23789689.2023.2233759

To link to this article: https://doi.org/10.1080/23789689.2023.2233759

SUSTAINABLE AND RESILIENT INFRASTRUCTURE
2023, VOL. 8, NO. 6, 572–588
https://doi.org/10.1080/23789689.2023.2233759

Taylor & Francis
Taylor & Francis Group

Predicting tree failure likelihood for utility risk mitigation via a convolutional neural network

Atanas Apostolov [a], Jimi Oke [a], Ryan Suttle [b], Sanjay Arwade [a] and Brian Kane [b]

[a]Department of Civil and Environmental Engineering, University of Massachusetts Amherst, Massachusetts, USA; [b]Department of Environmental Conservation, University of Massachusetts Amherst, Massachusetts, USA

ABSTRACT

Critical to the resilience of utility power lines, tree failure assessments have historically been performed via costly manual inspections. In this paper, we develop a convolutional neural network (CNN) to predict tree failure likelihood categories (*Probable*, *Possible*, *Improbable*) under three classification strategies. The CNN produced the best performance under the *Probable*/*Possible* vs. *Improbable* strategy, achieving a recall score of 0.82. We also perform a visual analysis of the predictions via Grad-CAM++ heatmaps, indicating an approach for incorporating interpretability into model selection. Benchmarking the results of our model against those produced by two state-of-the-art CNNs (ResNet-50 and Inception-v3), we show that our relatively simple model produces better results in a computational time that is three times faster. Via this novel framework, we demonstrate the potential of artificial intelligence to automate and consequently reduce the costs of tree failure likelihood assessments in proximity to power lines, thereby promoting sustainable infrastructure.

ARTICLE HISTORY
Received 7 September 2022
Accepted 13 June 2023

KEYWORDS
Utility power lines; CNN; AI; infrastructure management; resilience

1. Introduction

Contacts between tree parts and power lines cause outages that annually result in tens of billions of dollars for maintenance and repair throughout the United States, despite extensive efforts by utilities to prevent them. Contact between tree parts and power lines can take three forms: tree branches can grow into lines; branches can fail and fall onto lines; or whole-tree failure can occur due to uprooting or trunk failure. A study in Connecticut, U.S.A., documenting annual disruptions of $8.3 billion between 2005 and 2015, provides some context for the amount of economic disruption (Graziano, Gunther, Gallaher, et al., 2020). This extremely high cost occurred despite extensive efforts on the part of utilities to mitigate conflicts between trees and power lines via active and aggressive pruning programs that, on their own cost billions of dollars annually (Guggenmoos, 2003).

Presently, the process of identifying high-risk trees near power lines is labor-intensive and time-consuming, as it requires visual and quantitative assessments by qualified personnel. Despite its high cost, pruning trees to maintain clearance from power lines is an effective way to reduce outages due to so-called 'preventable' contacts between trees and power lines. For example, in Massachusetts, U.S.A., where tree failure was responsible for 40% of preventable tree-caused outages, pruning was able to improve reliability by 20% to 30% (Simpson & Van Bossuyt, 1996). Similar results were found in a study conducted in Connecticut (Parent, Meyer, Volin, et al., 2019). The efficacy of pruning has also been demonstrated in a study of two states in the US Gulf Coast region, where the inclusion of pruning resulted in reduced uncertainty in the prediction of wind-induced power outage models (Nateghi, Guikema, & Quiring, 2014).

Yet, effective pruning cannot completely eliminate tree-caused outages, as they can still occur when high-risk trees – even those outside the right-of-way – fail and impact power lines in contact (Guggenmoos, 2003). For instance, an estimated 95% of tree-caused outages in the Pacific Northwest region of the U.S.A. were due to tree failure (Guggenmoos, 2011). As another example, 25% of interruptions in Illinois, U.S.A., were reportedly due to trees that uprooted or broke in the stem (Wismer, 2018). Thus, the capability to assess tree risk can facilitate targeted pruning efforts in order to further improve resilience.

While an inexact science, tree risk assessment best management practices have been developed over the years (Goodfellow, 2020; Smiley, Matheny, & Lilly,

CONTACT Atanas Apostolov aapostolov@umass.edu

2017). Estimating tree risk includes the assessment of: likelihood of tree failure; likelihood of impact of the failed tree (or tree part) on a target; and severity of consequences of the impact. The likelihood of failure depends on the anticipated loads on the tree and its load-bearing capacity. The likelihood of impact depends on proximity to the target (lines, poles, and other hardware—"infrastructure" – in the case of utility tree risk assessment), the target's occupancy rate (which is constant for utility lines) and whether the target is protected, for example by neighboring trees. The severity of consequences depends on the damage done to the infrastructure – which, in turn, is partially related to the size of the tree or tree part that fails, and how much momentum it has when it impacts the infrastructure – and, more importantly in some cases, the economic costs and disruption associated with electrical outages.

Individual tree risk assessment can be costly because of the time it requires. In some situations, a less time-consuming assessment may be justified to reduce costs, i.e., a 'Level 1' assessment (Smiley, Matheny, & Lilly, 2017). Studies in Rhode Island, U.S.A. (Rooney, Ryan, Bloniarz, et al., 2005) and Florida, U.S.A. (Koeser, McLean, Hasing, et al., 2016) have shown that, compared to more time-consuming risk assessments, Level 1 risk assessments successfully identified trees with a higher degree of risk – precisely the trees that arborists prioritize for risk mitigation. The utility of Level 1 assessments demonstrated in these studies suggests that artificial intelligence (AI) tools may be an effective way to reduce the cost of tree risk assessment while still identifying high-risk trees.

At the heart of AI-based visual analytic framework lies the convolutional neural network (CNN), which can be trained to perform tasks such as image classification, object and text detection, document tracking, among others (Gu, Wang, Kuen, et al., 2018). The groundbreaking study of Hubel and Wiesel (1959) showed that visual perception in cats was a result of the activation or inhibition of groups of cells in the visual cortex known as 'receptive fields.' Further, they attempted to map the cortical architecture in cats and monkeys (Hubel & Wiesel, 1962, 1965, 1968). Subsequent attempts were then made to model neural networks that could be trained to automatically recognize visual patterns with modest performance (Fukushima, 1975; Giebel, 1971; Kabrisky, 1966; Rosenblatt, 1962). However, the breakthrough in artificial intelligence (AI) for image recognition came with the 'neocognitron' (Fukushima, 1980), which was a self-learning neural network for pattern recognition that was robust to changes in position and shape distortion, a problem that plagued earlier efforts, including the 'cognitron' also proposed by Fukushima (1975).

A few notable efforts demonstrated that neural networks were viable for handwritten digit recognition (Denker, Gardner, Graf, et al., 1988; Fukushima, 1988), but these required significant preprocessing and feature extraction. LeCun, Boser, Denker, et al. (1989) soon afterward introduced a multilayer neural network that mapped a feature in each neuron (representing a 'local receptive field') via convolution. This network could also be trained by backpropagation like other existing neural networks and featured pooling operations for better distortion and translation invariance. Further developments from this milestone yielded the LeNet-5 CNN, which attained accuracy levels that rendered it commercially viable (Lecun, Bottou, Bengio, et al., 1998).

Successive breakthroughs have yielded high-performing CNN architectures exploiting advancements in image capture capabilities, big data and computation. AlexNet (Krizhevsky, Sutskever, & Hinton, 2012), with five convolutional layers and three dense layers-one of the largest CNNs of its time – won the 2012 ImageNet Large Scale Visual Recognition Challenge (ILSVRC-2012; held annually from 2010 through 2017) with a landmark top-5 error rate of 15.3%. Zeiler and Fergus (2014) then introduced ZFNet, besting the performance of AlexNet. They also pioneered visualization techniques that were foundational for model inference and interpretability. Szegedy, Liu, Jia, et al. (2014) then introduced GoogLeNet, a 22-layer network, featuring the novel 'Inception module,' which paved the way for efficiency in a deep network by concatenating outputs from multiple filter sizes in a single module. Subsequent improvements have since been implemented on the original Inception framework (Szegedy, Ioffe, Vanhoucke, et al., 2016; Szegedy, Vanhoucke, Ioffe, et al., 2015). VGGNet (Simonyan & Zisserman, 2015) also pushed the boundaries of depth with up to 19 weight layers, achieving state-of-the-art performance at ILSVRC-2014. ResNet (He, Zhang, Ren, et al., 2015) then addressed the accuracy degradation problem arising from increasing depth in a network by successively fitting smaller sets of layers to the residual and employing skip connections. This advance led to unprecedented levels of depth, as ResNets with 34, 50, 101 and 152 layers were demonstrated, with ResNet-152 winning first place in ILSVRC-2015. Innovations in cardinality resulted in CNNs that efficiently utilize model parameters for improved performance, such as ResNeXt (Xie, Girshick, Dollár, et al., 2017) and Xception (Chollet, 2017). Attention-based developments, such as the Residual Attention Network (RAN) (F. Wang, Jiang, Qian, et al., 2017) and the Convolutional Block Attention Module (CBAM)

(Woo, Park, Lee, et al., 2018), further increased interpretability and performance. Details of more recent CNN advances are provided by Alzubaidi, Zhang, Humaidi, et al. (2021).

Given these innovations, the application of AI to infrastructure management is expected to continue to grow. Concurrently, the increasing digitalization of infrastructure should spur the development and operation of sustainable systems and provide richer datasets for better AI tools (Donti & Kolter, 2021; McMillan & Varga, 2022). In particular, the potential for AI to facilitate effective risk assessment has been demonstrated in applications ranging from earthquakes (Jiao & Alavi, 2020; Salehi & Burgueño, 2018) and structural health (Spencer, Hoskere, & Narazaki, 2019; N. Wang, Zhao, Wang, et al., 2019) to landslides (Su, Chow, Tan, et al., 2020), vegetation (dos Santos, Marcato Junior, Araújo, et al., 2019; Egli & Höpke, 2020; Fricker, Ventura, Wolf, et al., 2019) and identifying critical infrastructure (Elliott, Shields, Klaehn, et al., 2022), among others. Yet, the development and adoption of AI tools for tree-utility risk assessment has been relatively slower (Matikainen, Lehtomäki, Ahokas, et al., 2016; Nguyen, Jenssen, & Roverso, 2018). This is due in part to the greater levels of accuracy required and the complicating factors specific to this problem, such as the large number of tree species to be considered; seasonal variation in tree appearance and associated risk; and local meteorological conditions.

Current research efforts to deploy AI for tree-utility risk assessment have focused largely on object detection. For instance, Zhang, Yuan, Li, et al. (2017) proposed an automated power line inspection framework that computes distances from vegetation. Watanabe, Ren, Zhao, et al. (2018) then classified sub-boxes of an image as 'tree,' 'power line' or 'other.' Conflicts were identified when a sub-box classified as 'tree' was surrounded by 'power line' sub-boxes or vice-versa. Similarly, Oliveira, Buckeridge, and Hirata (2022) proposed a novel approach of detecting trees entangled with power and communication lines via street-level imagery. In a more relevant effort, Gazzea, Pacevicius, Dammann, et al. (2021) developed an AI framework that uses a CNN to detect four classes of tree density in a power grid, with greater density considered as posing greater risk to the utility lines.

However, there remains a gap in deploying AI for predicting the risk of actual tree failure in proximity to power lines, which is critical to the ability of utilities to better direct pruning efforts for improved resilience and at lower costs. In this study, we demonstrate that a relatively simple CNN architecture based on state-of-the-art approaches for model training and regularization can effectively address this problem, and with further development, emerge as a viable tool for real-time applications. Our contribution is thus an efficient and cost-effective AI approach via a CNN to classify smartphone camera images of trees among three categories of failure likelihood: *Improbable*, *Possible*, and *Probable*. The data used for training, testing and illustration of the method consists of 933 tree images that have been classified by the authors according to best management practices used by utility arborists (Goodfellow, 2020). With further development, this approach can be harnessed to reduce power outages and repair costs by allowing utilities to more effectively target their pruning and mitigation efforts. This potential can be realized, for instance, by coupling the AI tool with car-mounted cameras, drones or open-source street imagery. Risk predictions could then be obtained with fewer to no expert assessment hours required, thus leading to cost savings.

2. Methods

2.1. Data

The training dataset consisted of 933 images, each having an original size of 3024 × 4032 pixels. Images were captured over three field seasons in Massachusetts, U.S.A., between May 2020 and April 2022 and were limited to in-leaf trees to avoid any potential influence of changes in tree appearance due to seasonal leaf senescence on image processing. Locations were randomly selected across the state. Field assessments of trees to classify likelihood of failure followed the 'Level 1' methods outlined in the second edition of the International Society of Arboriculture's (ISA) Tree Risk Assessment Best Management Practices (Smiley, Matheny, & Lilly, 2017) and ISA's Utility Tree Risk Assessment Best Management Practices (Goodfellow, 2020). This method is commonly used to assess trees in the United States. The subjective nature of visual assessments introduces potential for inter-rater variation in data collection. Thus, four inter-rater reliability tests were conducted to evaluate consistency in data gathering. In each test, the assessors considered the same group of trees independently of one another on the same day, and data were compared to examine discrepancies. In a Level 1 assessment, one or two parameters in the risk assessment are typically held constant. In our study, we assumed the same levels of likelihood of impact and severity of consequences because the target in every risk assessment was the utility wires. In the field, the assessor examined the tree from the street side to assess its likelihood of failure.

A Level 1 assessment was selected for this study because: (i) individual risk assessments may be prohibitively expensive at higher orders, i.e., Level 2 or Level 3 (Smiley, Matheny, & Lilly, 2017), given the hundreds of thousands of trees utilities must manage across territory areas; (ii) utility right-of-way (ROW) easements may not allow utility inspectors full access to trees in practical application of higher order risk assessment procedure if the trees are beyond the edge of the ROW (Goodfellow, 2020); and (iii) studies have shown reasonable efficacy of limited basic visual assessment techniques in identifying more severe tree defects (Koeser, McLean, Hasing, et al., 2016; Rooney, Ryan, Bloniarz, et al., 2005) leading to greater likelihood of failure ratings. The four categories of likelihood of tree failure, which are always considered in a stated time frame, are defined as follows (Smiley, Matheny, & Lilly, 2017):

- *Improbable*: The tree or tree part is not likely to fail during normal weather conditions and may not fail in extreme weather conditions within the specified time frame.
- *Possible*: Failure may be expected in extreme weather conditions, but it is unlikely during normal weather conditions within the specified time frame.
- *Probable*: Failure may be expected under normal weather conditions within the specified 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 people from harm. The imminent category overrides the stated time frame.

In this study, our time frame is 3 years and 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. Trees in Figure 1a were categorized as *Improbable* due to their lack of structural defects as well as good physiological health. Trees in Figure 1b were categorized as *Possible* due to weak branch unions and crown dieback. Trees in Figure 1c were categorized as *Probable* because they were dead (above) or had a large visible crack through the trunk (below). In the original set of training images, the class distribution is given in Table 1.

(a) Improbable (b) Possible (c) Probable

Figure 1. Examples of training images in each of the three tree risk categories considered in this study.

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

Category	Number of images
Probable	171
Possible	207
Improbable	555
Total	933

Table 2. Classification strategies.

Strategy	Description	No. classes
Pr_Im	{*Probable*, *Improbable*}	2
PrPo_Im	{*Probable* + *Possible*, *Improbable*}	2
Pr_PoIm	{*Probable*, *Possible* + *Improbable*}	2

2.2. Classification strategies

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

Strategy 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, Thomas Smiley, Hauer, et al., 2020), in strategy PrPo_Im, we pooled trees in the *Probable* and *Possible* categories and compared them to trees in the *Improbable* category. In strategy Pr_PoIm, we pooled trees in the *Possible* and *Improbable* categories and compared them to trees in the *Probable* category. In practice, this strategy is less likely because arborists typically distinguish trees with an *Improbable* likelihood of failure as those with minimal or no structural defects. It requires additional judgment to distinguish trees with probable or possible likelihood of failure because an arborist must assess the severity of structural defects, the presence of response growth, and the expected loads (Smiley, Matheny, & Lilly, 2017).

2.3. Data pre-processing, augmentation and partitioning

In order to achieve computational efficiency, we downsampled the 933 input images from the original resolution of 3024×4032px to 336×336px. First, we randomly sampled 10% of all images and reserved these as the test set for final model assessment in order to ensure that performance estimates were based on observations previously unseen by the model. The remaining images were then partitioned into a training set and a validation set. Given the relatively small size of the data set, we apply the data augmentation techniques (Shorten & Khoshgoftaar, 2019; Wong, Gatt, Stamatescu, et al., 2016) of horizontal flipping to generate 'mirror images' from observations in the training and validation sets. Thus, we effectively doubled their size. We maintain a training-validation-test ratio of roughly 80:10:10 by accounting for the doubling of the training and validation set sizes via the following equation: $test_{size} = 2\alpha n/(1+\alpha)$, where α is the desired test size ratio and n is the total number of images. In this case, $\alpha = 0.1$. We show the exact number of images per set for each strategy in Table 3. During hyperparameter optimization, the training set was used for each set of candidate parameters, while metrics on the validation set were used for selection. For the final post-optimization training, the training and validation sets were combined into a single training set. The non-augmented 170-image test set, left untouched in all training procedures, was then used to generate performance metrics for final model assessments.

Table 3. Training/Validation/Test post-augmentation image split for each classification strategy.

Image Set	Pr_Im	PrPo_Im	Pr_PoIm
Training	1050 (80%)	1374 (81%)	1374 (81%)
Validation	132 (10%)	152 (9%)	152 (9%)
Test	135 (10%)	170 (10%)	170 (10%)
Total	1317 (100%)	1696 (100%)	1696 (100%)

As an example, Figure 2 shows pre-processed images downsampled to 336-px, with some horizontally flipped. These are organized according to the three classification strategies described.

2.4. Convolutional neural network

We employ a convolutional neural network (CNN) as the AI framework for tree risk failure likelihood prediction. Like other neural networks, the CNN is an arrangement of neurons within layers, each neuron performing an operation that maps from a pixel in an input image to the final output. The input into each neuron is a weighted sum from the previous layer, while the output from each neuron is modulated by an activation function. The activation function in the final layer of a CNN is typically the softmax function shown in Equation 1, which gives the class probabilities of a given input image. A class is assigned to the image based on the one with the corresponding maximum probability.

Unlike other neural networks, however, the CNN performs fundamental pixel-mapping operations 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 respectively-sized local receptive

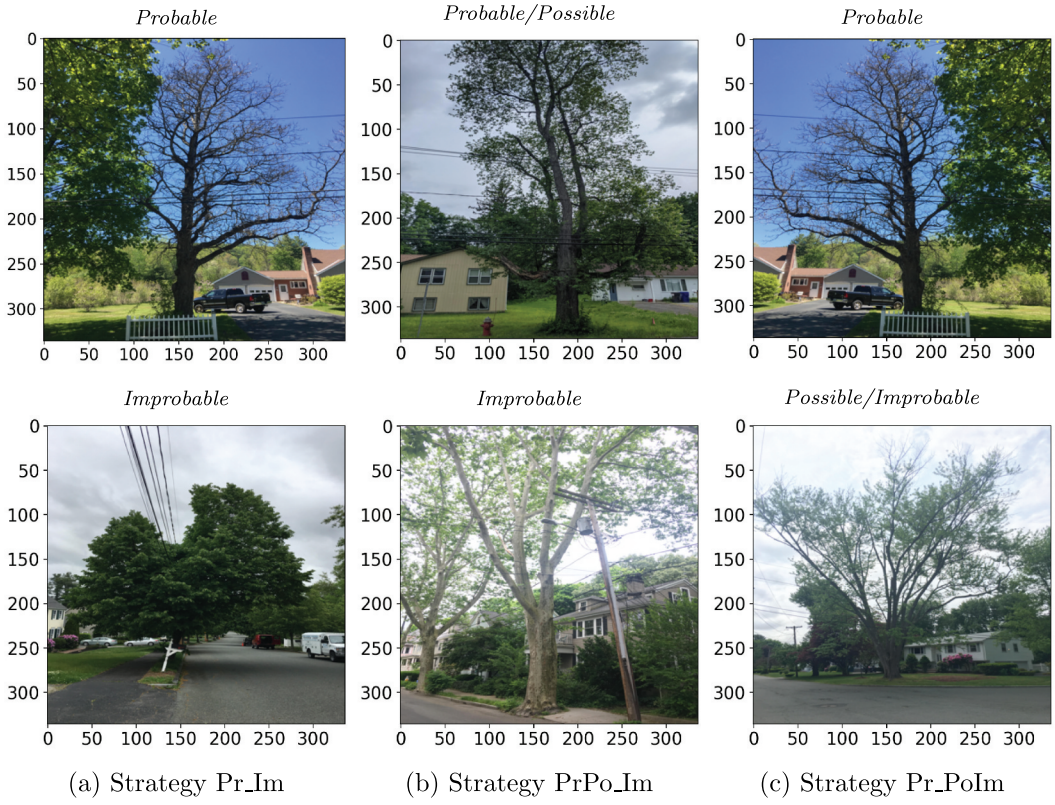

Figure 2. Selected processed training images under each classification strategy. Each has a resolution of 336×336 pixels, and some have been horizontally flipped.

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.

Here, we employ a relatively simple CNN architecture, historically inspired by AlexNet (Krizhevsky, Sutskever, & Hinton, 2012). The structure of the CNN is shown in Figure 3, generated using an automated framework (Bäuerle, van Onzenoodt, & Ropinski, 2021). We used five convolutional layers for our CNN as a trade-off between complexity and performance, given the relatively small number of images in our dataset, and also based on prior experimentation. Following the input layer (a matrix of pixels from the input image), we use a 64-filter convolutional layer. It has been empirically shown that CNNs perform best when the convolutional layers are arranged to capture increasing levels of abstraction from a given input image (He, Zhang, Ren, et al., 2015; Simonyan & Zisserman, 2015; Szegedy, Liu, Jia, et al., 2014). The level of abstraction is indicated by a combination of features (each one being captured by a filter). Thus, the first convolutional layer typically has a smaller number of filters, which increases with successive layers. Given the selected resolution of our input images and also based on initial experiments with 32, 64 and 128 filters, we selected 64 as the best choice. Powers of 2 were used historically for the sake of parallel computational efficiency and currently retained by convention. The output is

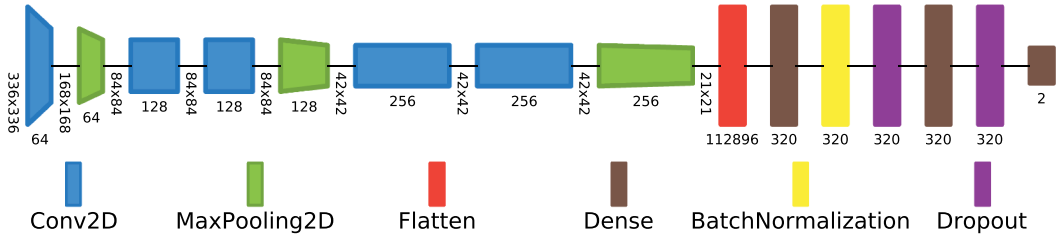

Figure 3. Diagram of the convolutional neural network structure (excluding the input layer). Hyperparameters that are optimized include the number of units in the penultimate dense layers. Here, 320 units are used, respectively.

Table 4. Summary of the convolutional neural network hyperparameters. Those represented by a symbol are optimized using a guided search.

Layer	Layer No.	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	

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 to a specified number of units. Batch normalization (Ioffe & 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' (Brigato & Iocchi, 2020) is used after each dense layer. In each dropout layer, a proportion of the neurons is randomly zeroed during training in order to improve the robustness of the model. The various hyperparameters in the model are summarized in Table 4.

2.5. Hyperparameter optimization

We optimized eight of the CNN hyperparameters using Hyperband (Li, Jamieson, DeSalvo, et al., 2018), an efficient guided grid-search algorithm. Searches were conducted for each classification strategy model, with the objective as the validation F_1 score. 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 5. For the kernel size in the first convolutional layer, we allowed for a choice between a 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, Bordes, & Bengio, 2011). Nevertheless, the tanh function remains a viable option. The dropout rates were allowed to range from 0 to 0.5 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 $[10^{-5}, 10^{-3}]$. The data used for the hyperparameter optimization procedure were strictly from the augmented training dataset, with 10% randomly sampled for validation (Table 3). The earlier reserved portion of images from the original dataset remained untouched until final model assessment.

Table 5. Optimal hyperparameters found using the Hyperband search algorithm for the three classification strategies.

	Classification Strategy		
Parameter	Pr_Im	PrPo_Im	Pr_PoIm
1st conv. kernel size, k	7	5	7
1st dense activation, a_1	ReLU	tanh	ReLU
2nd dense activation, a_2	tanh	ReLU	tanh
1st dropout rate, r_1	0.25	0.25	0.35
2nd dropout rate, r_2	0.3	0.35	0.35
1st dense layer units, u_1	192	416	160
2nd dense layer units, u_2	352	320	160
learning rate, λ ($\times 10^{-5}$)	1.76	1.09	1.4

2.6. Model development

In this subsection, we provide an overview of the learning procedure for the convolutional neural network. All the symbols used here are summarized in Table 6.

2.6.1. Training

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:

Table 6. Summary of symbols related to the training and assessment of the convolutional neural network.

Symbol	Definition
c	Index of given class
$f(s^c)$	Softmax activation function
F_1	F_1 score (harmonic mean of precision and recall)
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 ith observation belongs to class c
Pr	Precision
Re	Recall
s^c	Class-specific score
y_i	Observed (true) class of a given observation
$\hat{y}_i$	Predicted class of a given observation
w_l^c	Gradient-based weight for feature map l (Original GradCAM)
w_l^{++c}	Gradient-based weight for feature map l (GradCAM++)
A^l	Activation of feature map l
j	Index of the width dimension for feature map A
k	Index of the height dimension for feature map A
Z	Number of pixels in the feature map
$GC_{j,k}^c$	GradCAM heatmap distinguishing features for predicted class c
$GC_{j,k}^{++c}$	GradCAM++ heatmap distinguishing features for predicted class c

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

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. We trained the CNN using a robust and adaptive variant of the stochastic gradient algorithm, Adam (Kingma & Ba, 2017), which computes adaptive learning rates for each weight parameter in the network. The goal of the training procedure is to learn the optimal weights and bias terms for the CNN by minimizing a loss function. In this case, we used the categorical cross-entropy loss function, which for a single observation can be simply defined as:

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

Training is iteratively performed, with the gradient of the loss function computed and averaged over a batch of input images. Here, we used a batch size of 32. While the Adam optimizer is parameter-wise adaptive, it still requires an initial learning rate, the choice of which could affect training performance. We optimized for this in the hyperparameter search as discussed. Furthermore, the CNN is trained over multiple passes through the entire training set. Each such pass is referred to as an epoch. In order to avoid overfitting, we employed early stopping to determine the stopping point in each model training instance. Given an objective performance metric, early stopping monitors the metric at the end of each epoch and tracks where its optimal value is attained. If there is no improvement in the optimal value after a specified number of epochs (known as the 'patience'), model training is terminated and weights are reverted to their optimal point.

2.6.2. Performance metrics

In real terms, we measured the performance of the trained CNN by how accurately it predicts the positive class in a validation or test set excluded from the training set. We used a randomly sampled validation set in each training instance during hyperparameter optimization. For the final strategy assessment under the optimized parameters, we trained the model using the entire training set and then computed performance metrics using the test set.

We define the accuracy, Acc, as the overall proportion of correct predictions across all classes. This metric was computed both for the training and test sets in each epoch. In addition to accuracy, we assessed the models trained under these strategies based on the precision, recall and F_1 score metrics computed over the test set in each epoch and considered for the positive class, as the one of interest. The precision score, Pr, captures the proportion of correct positive predictions relative to all the positive predictions, and is an important measure of how good a classifier is. The recall, Re, captures the ability of a classifier to correctly predict the observed positives. The F_1 metric is given as the harmonic mean of the precision and recall, and is thus more sensitive than the overall accuracy score. These metrics are formally defined as follows:

$$\text{Accuracy}: Acc = \frac{TP + TN}{TP + FP + FN + TN} \tag{3}$$

$$\text{Precision}: Pr = \frac{TP}{TP + FP} \tag{4}$$

$$\text{Recall}: Re = \frac{TP}{TP + FN} \tag{5}$$

$$\text{F}_1\text{score}: F_1 = \frac{2 \cdot Pr \cdot Re}{Pr + Re} \tag{6}$$

where

$$\text{True positives}: TP = \sum_i \mathbb{I}\left(\hat{y}_i = y_i = 1\right) \tag{7}$$

$$\text{True negatives}: TN = \sum_i \mathbb{I}\left(\hat{y}_i = y_i = 0\right) \tag{8}$$

$$\text{False positives}: FP = \sum_i \mathbb{I}\left(\hat{y}_i \neq y_i = 1\right) \tag{9}$$

$$\text{False negatives}: FN = \sum_i \mathbb{I}\left(\hat{y}_i \neq y_i = 0\right) \tag{10}$$

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

2.6.3. Gradient-weighted class activation maps

Interpretability is a critical concept for understanding the predictions generated by a deep model beyond the traditional performance metrics. It promotes transparency by providing a rationale behind the returned output. The gradient-weighted class activation maps (Grad-CAM) is a highly class-discriminative localization approach (Chattopadhyay, Sarkar, Howlader, et al., 2018; Selvaraju, Cogswell, Das, et al., 2020) which offers a visual explanation of the model by highlighting regions in an image deemed relevant in the prediction process. A generalization of the Class Activation Map (CAM) method (Zhou, Khosla, Lapedriza, et al., 2016), the Grad-CAM is applicable to a broad range of CNNs without the need for architectural changes. Based on the predicted class, the Grad-CAM heatmap facilitates a visual assessment of model explainability. The class-specific Grad-CAM heatmap image $GC^c_{j,k}$ is given by:

$$GC^c_{j,k} = \text{ReLU}\left(\sum_l w^c_l A^l_{j,k}\right) = \max\left\{0, \sum_l w^c_l A^l_{j,k}\right\} \tag{11}$$

where j and k are the indices for the width and height of the input image and $A^l_{j,k}$ is the activation (matrix) of feature map l. The gradient-based weight, w^c_l is given by:

$$w^c_l = \frac{1}{Z}\sum_j \sum_k \frac{\partial s^c}{\partial A^l_{jk}}, \quad j = 0, 1, \ldots, M-1; k = 0, 1, \ldots, N-1 \tag{12}$$

where M is the height and N is the width of the image and $Z = M \times N$ is the number of pixels in each feature map.

GradCAM++, an enhanced implementation of the original GradCAM, has demonstrated greater discriminative capability (Chattopadhyay, Sarkar, Howlader, et al., 2018). Thus, we use GradCAM++ instead to produce the attention heatmap for visually analyzing the CNN predictions. It is similarly given by:

$$GC^{++c}_{j,k} = \text{ReLU}\left(\sum_l w^{++c}_l A^l_{j,k}\right) \tag{13}$$

The difference lies in the weighting approach, as the activation weight w^{++c}_l is redefined to retain more of the structural properties of the feature maps and are given by:

$$w^{++c}_l = \sum_j \sum_k \left[\frac{\frac{\delta^2 s^c}{\delta(A^l_{j,k})^2}}{2\frac{\delta^2 s^c}{\delta(A^l_{j,k})^2} + \sum_a \sum_b A^l_{a,b} \frac{\delta^3 s^c}{\delta(A^l_{j,k})^3}}\right] \tag{14}$$

The heatmap generated by GradCAM++ is upsampled to the resolution of the training image and superimposed on the original for assessment. After training the CNN for each of the classification strategies and generating test performance metrics, we analyze the prediction probabilities to gain insight into the certainty of the model for the best performing strategy. Furthermore, we perform a visual assessment of model interpretability by comparing Grad-CAM++ images at various prediction probabilities, for true positives, true negatives, false positives and false negatives. In future research, the explainability of the Grad-CAMs can be quantified in order to use this as an objective for model selection.

3. Results and discussion

3.1. Model performance

We trained the CNN for each of the three classification strategies, namely, Pr_Im, PrPo_Im and Pr_PoIm with the goal of analyzing their effectiveness for tree failure risk assessment. As earlier noted, the F_1 score is more sensitive to the ability of the classifier to distinguish between classes, as it incorporates both the precision and recall. For practical applications, higher F_1 (and, consequently, Pr and Re) scores are desired in order to have classifiers that reasonably predict classes that

include higher-risk categories than *Improbable*, which is the most represented category in this study. Thus, we used the F_1 metric as the early stopping monitor, with a patience of 6 epochs, to find the optimal stopping point of the CNN in each training instance. We evaluated model performance using the test set yet-unseen by the model. We show the trajectories of the performance metrics for single training instances in each strategy in Figure 4. Pr_Im is optimal with respect to F_1 from the first epoch (hence, only seven epochs were required for training). For PrPo_Im the optimal epoch was 10. From Figure 4, we see that while Pr_PoIm attains the greatest test accuracy scores, it is not as useful for risk assessment, given its lower F_1 score. Pr_Im, which produces the next highest test accuracy score, is not effective in predicting *Probable* trees when it is most accurate, given that its F_1 score is less than 0.5. This behavior further motivates the use of the F_1 (which is the average of the harmonic mean of precision and recall) to track the classifier performance.

Corresponding confusion matrices at the optimal epoch for each strategy (based on F_1) are shown in Figure 5. The values in the top row of the confusion matrices indicate the TN and FP predictions, respectively, while those in the bottom row indicate the FN and TP predictions, respectively. The confusion matrices show that, generally, the *Probable* category is predicted most accurately as an individual class in the absence of *Possible* images (Pr_Im strategy). Meanwhile, the classifier generally performed best at recalling the class containing the *Improbable* category when it was grouped with *Possible*. These observations align with previous work (Koeser, Thomas Smiley, Hauer, et al., 2020) and are 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,

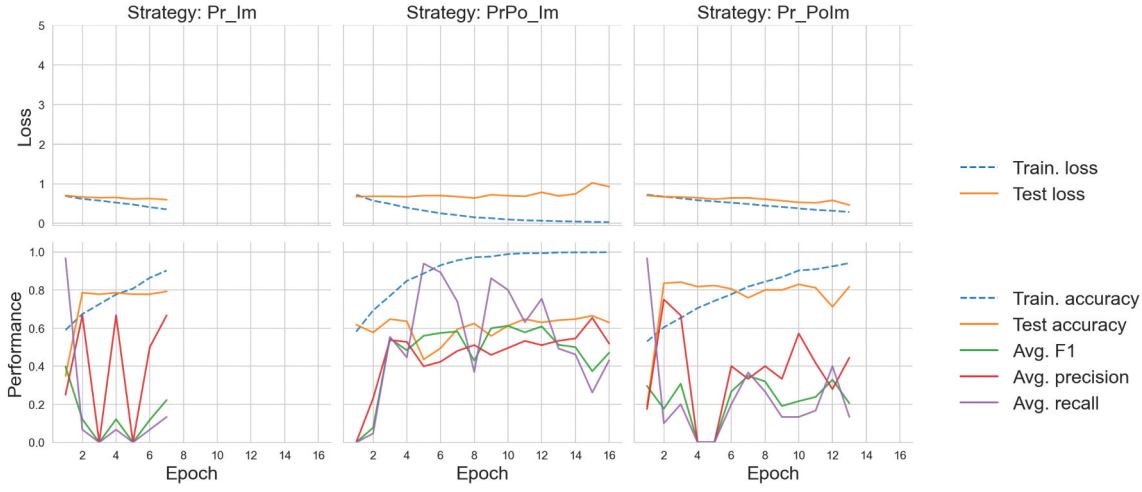

Figure 4. Model performance across three classification strategies for a single training instance in each case. All weighted average metrics were computed on the test set.

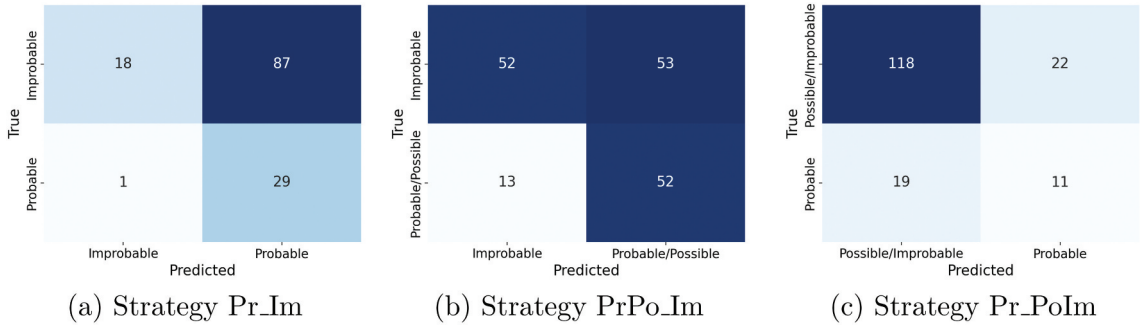

Figure 5. Confusion matrices for a single training instance for each classification strategy. Top left indicates the TN, top right indicates the FP, bottom left indicates the FN, and bottom right indicates the TP predictions.

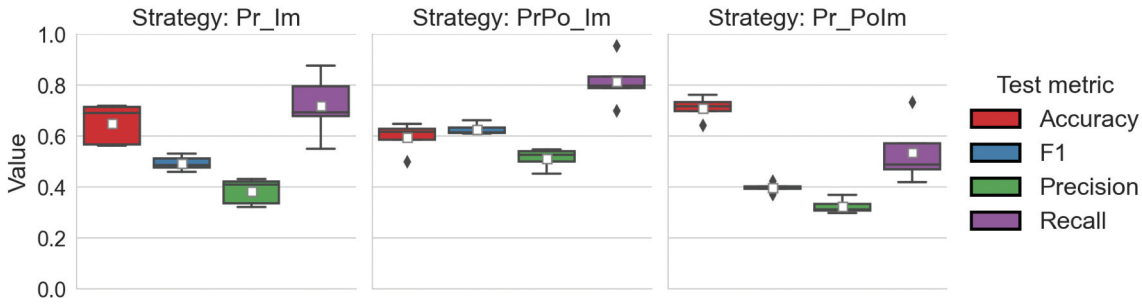

Figure 6. Boxplots of test performance metrics for each classification strategy.

and the anticipated loads on the tree. Although there are guidelines to inform an arborist's assessment (Goodfellow, 2020; Smiley, Matheny, & Lilly, 2017), 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 the classifier's ability to recall *Probable* and *Possible* trees was better when both categories were grouped into the same class (Figure 5b) than when *Possible* and *Improbable* were in the same group. Therefore, we would select the PrPo_Im strategy as the best strategy from a practical standpoint.

To obtain stable overall performance estimates, we trained the model five times under each strategy and averaged the metrics at the optimal stopping point in each case. Figure 6 shows the boxplots of the validation performance metrics (accuracy, precision, recall and F_1) for the three cases considered. 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 indicated by square markers. From Figure 6, we observe that while Pr_Im and Pr_PoIm produce the highest accuracy scores (mean = 0.65 and mean = 0.71, respectively), their corresponding optimal F_1 values are lower compared to PrPo_Im.

We further summarize the statistics of the test performance metrics in Table 7. In practice, the Pr_Im strategy might be considered a less relevant strategy, as *Possible* trees cannot be avoided altogether. Of the three strategies, PrPo_Im has the best F_1 score of 0.63. Even though this strategy had a lower accuracy score (0.6) than the Pr_PoIm strategy (0.71), it is the most useful in practice, as the ability to predict higher-risk categories (*Probable* and *Possible*) can be considered more useful than predicting the *Improbable* category from a risk-mitigation perspective. Furthermore, PrPo_Im had the highest average recall of 0.82. These results show that it is most viable to group *Probable* and *Possible* into the same class in order to have the best recall and precision performances from the CNN.

3.2. Classifier certainty and visual interpretation

We analyze the prediction probabilities on the test set across four categories: true positives (TP)—*Probable/Possible* images that are correctly predicted; false positives (FP)—*Improbable* images that are incorrectly predicted; true negatives (TN)—*Improbable* images that are correctly predicted; and false negatives (FN)—*Probable/Possible* images that are incorrectly predicted. We use violin plots to show the distributions of the probabilities (Figure 7). Summary statistics are provided in Table 8. We observe that the model is most certain (mean prediction probability of 0.71) when correctly classifying the positive class (PrPo) in the PrPo_Im strategy, which aligns with what practitioners are primarily interested in from a risk management perspective. The next highest mean of prediction probabilities (0.67) belongs to the false positives. Both TN and FN average 0.64. We expect that as the discriminatory capability of a classifier improves, prediction probabilities for TP and TN should increase, while those for FP and FN should decrease.

Table 7. Mean and standard deviation of classifier performance estimates of the three strategies investigated over five model training instances. 'SD' represents 'standard deviation'.

	Acc		F_1		Re		Pr	
Strategy	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Pr_Im	0.65	0.08	0.49	0.03	0.72	0.12	0.38	0.05
PrPo_Im	0.60	0.06	0.63	0.02	0.82	0.09	0.51	0.04
Pr_PoIm	0.71	0.04	0.40	0.02	0.54	0.12	0.32	0.03

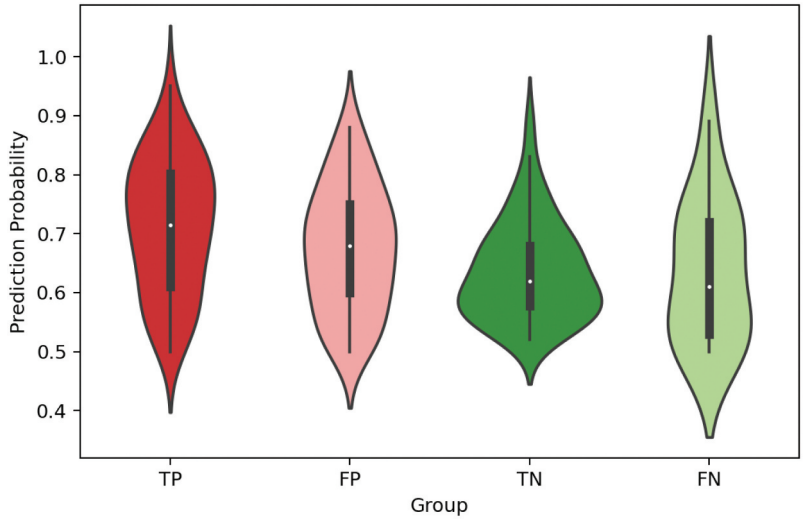

Figure 7. Violin plots indicating the distribution of probabilities for True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN) predictions in the PrPo_Im strategy.

Table 8. Summary statistics of true positive, false positive, true negative and false negative rates of the prediction probabilities.

	Mean	SD	Min	25th pctl(Q1)	50th pctl (Q2/median)	75th pctl (Q3)	100th pctl (Q4/max)
True Positive	0.71	0.11	0.50	0.61	0.72	0.80	0.95
False Positive	0.67	0.11	0.50	0.60	0.68	0.75	0.88
True Negative	0.64	0.08	0.52	0.58	0.62	0.68	0.89
False Negative	0.64	0.12	0.50	0.53	0.61	0.72	0.89

As an additional interpretative tool, Grad-CAM++ visualizations indicate what regions of an image the CNN focuses on when making predictions. We assess the correctness of the CNN in making its predictions by interpreting heatmaps which show these areas of attention. The more interpretable the CNN, the more consistently it should highlight the areas of interest (i.e., conflict regions) for the correct predictions. Similarly, we may expect to see that false positive (FP) images would incorrectly highlight non-conflict regions in the image. We generated Grad-CAM++ visualizations at the 25th, 50th, 75th and 100th percentiles of the prediction probabilities for selected TP, FP, TN and FN images (Figure 8). Based on these visualizations, we observe that the CNN tends to highlight tree parts visibly in contact with power lines (conflict regions) when classifying an image as *Probable/Possible*. The explainability of the attention areas increases with the prediction probability.

In the 75th and 100th percentiles for TP predictions, for instance, we see clearly that the Grad-CAM++ heatmap correctly highlights conflict regions (Figure 8a). Conversely, for TN predictions, the same percentiles indicate attention on the non-conflict regions. Specifically, these non-conflict regions are those that outline tree cover or utility infrastructure that is free of interaction (Figure 8c). In the FP examples (Figure 8b), we observe, however that the heatmap indicates an incorrect perception of conflict. For instance, in the 50th-percentile FP image, the clouds in the background of the power lines appears as tree cover to the CNN. In the 100th-percentile FP image, the issue appears to be incorrect depth perception. In reality, the power line is in the foreground, while the tree is in the background, and a house is even further back. Yet, the CNN 'sees' conflict between the tree and the power lines, and also between the tree and parts of the house. For the FN images (Figure 8d), the reverse case of depth perception is at play (50th and 75th percentiles), along with a non-identification of the conflict region (100th percentile).

Overall, however, these visual observations are promising. They indicate that with more training data, an interpretable and highly sensitive CNN can be achieved. The Grad-CAM++ further provides an avenue to quantify the amount of attention that can be correctly interpreted as conflict or non-conflict. This quantification can be then used as an objective in selecting from among competing models or classification strategies.

3.3. Comparison with state-of-the-art architectures

We benchmark the performance of our model (termed 'Base CNN' in this section) against that of two existing high-performance architectures, namely ResNet-50 (He,

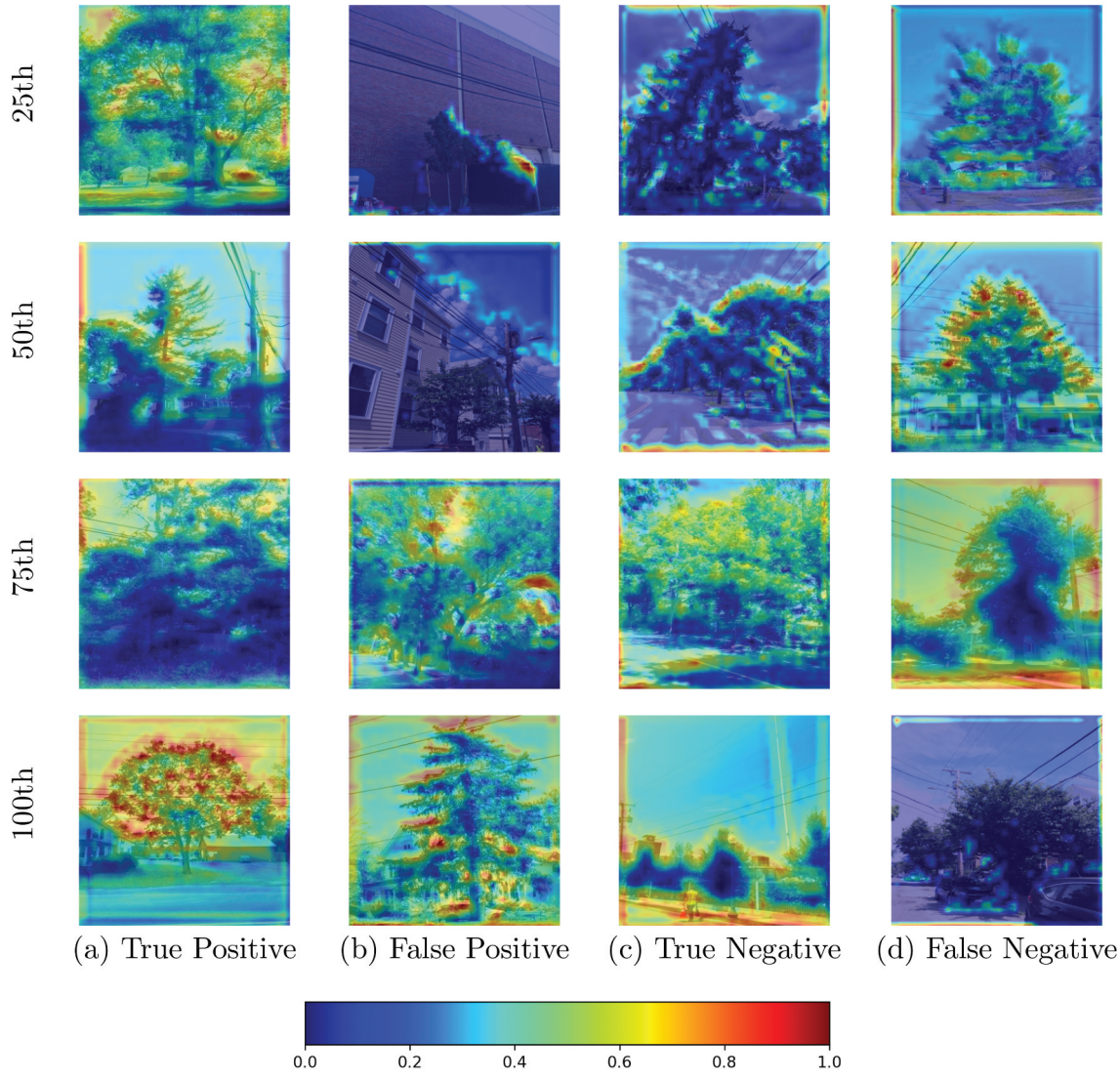

Figure 8. Grad-CAM++ visualizations for selected images in the PrPo_Im strategy in the four quartiles of prediction probability for true positives, false positives, true negatives and false negatives.

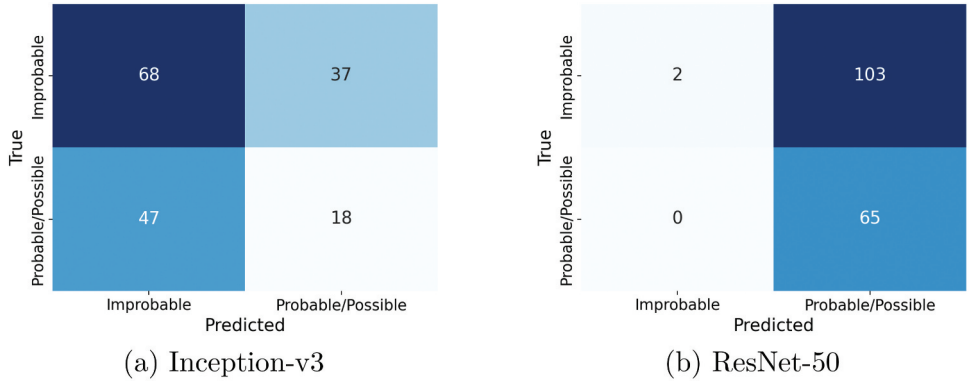

Figure 9. Confusion matrices for (a) Inception-v3 and (b) ResNet-50 trained under the PrPo_Im strategy..

Table 9. Predictive and computational performance of the state-of-the-art CNN architectures trained on 336 × 336 pixel input images under the classification strategy PrPo_Im.

	Test metrics				Execution performance		
Model	Accuracy	*Pr*	*Re*	F_1	Time/epoch (s)	No. epochs	Total time (h:m)
Base CNN	0.61	0.50	0.80	0.61	30	16	0:08
Inception-v3	0.51	0.33	0.28	0.30	153	9	0:23
ResNet-50	0.39	0.39	1.00	0.56	132	10	0:22

Zhang, Ren, et al., 2015), which has 50 convolutional layers, and Inception-v3 (Szegedy, Ioffe, Vanhoucke, et al., 2016), which consists of 48 layers. We focus our comparison on the PrPo_Im strategy, as this resulted in the best performance for the Base CNN, and also the most relevant for practical applications, as earlier discussed. The confusion matrices for Inception-v3 and ResNet-50 are presented in Figure 9. Predictive and computational performance metrics of all three models are summarized in Table 9.

In training the ResNet-50 and Inception-v3 models, we also used the Adam optimizer with an inital learning rate of 0.001. As with the Base CNN, we employed early stopping on these two models, as well, by monitoring the F_1 score with a patience of 6. While the Base CNN required 16 epochs for training, Inception-v3 and ResNet-50 required 9 and 10, respectively. Given the depth of these networks, they required more time to train per epoch than the Base model (both by a factor of 3). Taking into account the number of epochs required to determine the F_1-optimal state in each network, the total training time for Inception-v3 and ResNet-50 was approximately 25 min each. In contrast, the Base CNN we developed required less than 10 min to train. All models were trained on an 80-core Intel Xeon Gold 6242 R 3.10 GHz CPU with 782 GB of memory available.

In this training instance, the Base CNN resulted in the highest F_1 score of 0.61. The rest of the performance metrics of our model also exceeded those of the two state-of-the-art networks (Table 9). Inception-v3 produced a considerably worse recall score of 0.28, compared to the Base CNN's 0.8 (Table 9). While ResNet-50 perfectly predicted all the positives ($Re = 1$), its precision and accuracy scores were the lowest among all three models ($Acc = Pr = 0.39$).

In sum, in a single training instance, the relatively simple Base CNN model produced a classifier that was significantly more sensitive to the higher-risk tree category (*Probable/Possible*) compared to the state-of-the-art Inception-v3 model. It also outperformed both Inception-v3 and ResNet-50 in terms of accuracy, precision and F_1. Furthermore, the 5-layer architecture of our Base CNN model required considerably less time to train than the two deep networks we considered, which both have about 10 times as many layers. Moreover, the simplicity of the Base CNN can more readily facilitate visual inference for model interpretability and thus guide future improvements. One caveat in this comparison, however, is that the performances of the Inception-v3, ResNet-50, and even of the Base CNN model, can all be improved with considerably more training data samples and increased representation of the positive (*Probable/Possible*) class.

4. Conclusion

We have demonstrated the potential of an artificial intelligence framework to predict tree failure likelihood (*Probable*, *Possible* and *Improbable*) with respect to utility infrastructure. Prior related efforts have either focused on identifying vegetation proximity to utility infrastructure or identifying tree-power line conflicts. In this paper, we addressed the problem of assessing the failure risk of trees that are already proximal to power lines.

Specifically, we developed and trained a five-layer convolutional neural network. From an initial resolution of 3024 × 4032 pixels, we compressed the resolution of the input images to 336 × 336px. We defined three classification strategies to investigate the performance of various groupings of the three categories: Pr_Im (*Probable* vs. *Improbable*); PrPo_Im (*Probable* and *Possible* vs. *Improbable*); and Pr_PoIm (*Probable* vs. *Possible* and *Improbable*). We then optimized eight of our CNN hyperparameters via a Hyperband search. We trained the CNN under each of the three strategies, and assessed final model performance based on accuracy, precision, recall and F_1 on a 170-image test set, averaging results over five model training runs.

Our results indicated that the Base CNN performed best at recalling *Probable*/*Possible* vs. *Improbable*. We found that while the Pr_PoIm strategy resulted in the highest overall accuracy scores (0.71), it performed poorly in recalling the classes containing the higher-risk *Probable* and *Possible* categories. In this regard, the PrPo_Im produced the best performance, resulting in the best possible F_1 score of 0.63, and consequently the best precision and recall scores, as well. The best high-risk recall score of 0.82 was also achieved under

PrPo_Im. For practical applications, strategy PrPo_Im is also the most viable one, as it not only includes all three categories, but also allows for distinguishing the categories of greater relevance to utility risk (*Probable* and *Possible*).

We also compared the results of our Base CNN under the PrPo_Im strategy to those produced by two state-of-the-art models, namely Inception-v3 and ResNet-50. We employed early stopping in training all three models. While Inception-v3 and ResNet-50 took three times as long to train, neither outperformed the Base CNN. In fact, they resulted in a significantly worse F_1 score and accuracy for the *Probable*/*Possible* class, compared to that of the Base CNN. Thus, we demonstrate that a relatively simple CNN has the potential to perform this classification task at a lower computational cost.

Given the relatively small input dataset of original images, these results are extremely promising for future improvements. In order to better understand how the CNN is classifying each image, we will conduct extensive visual inference in further work, for instance, quantifying the attention areas produced by Grad-CAM++ as an interpretability objective for model selection. There is also a potential for mapping the visual cues learned by the CNN to physical relationships governing tree structure, in order to gain greater insights into tree failure processes. The automation for tree failure likelihood assessments can potentially supplement human decision-making for increased resilience and, in the future, reduce costs and improve reliability in tree assessments, thus leading to more sustainable communities.

Acknowledgments

The authors acknowledge the partial support of Eversource Energy in funding this work. We also thank the anonymous reviewers whose insightful comments considerably improved the quality of the manuscript.

Disclosure statement

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

Notes on contributors

Atanas Apostolov is a graduate research assistant in the Department of Civil and Environmental Engineering at the University of Massachusetts Amherst.

Jimi Oke is an Assistant Professor in the Department of Civil and Environmental Engineering at the University of Massachusetts Amherst.

Ryan Suttle is an ISA Board Certified Master Arborist and Utility Specialist. He obtained his MS in Arboriculture and Urban Forestry from the University of Massachusetts Amherst.

Sanjay Arwade is a Professor of Civil Engineering at the University of Masachusetts Amherst.

Brian Kane is the Massachusetts Arborists Association Professor of Arboriculture at the University of Massachusetts Amherst. He has been an ISA Certified Arborist since 1997.

ORCID

Atanas Apostolov http://orcid.org/0000-0002-7321-6051
Jimi Oke http://orcid.org/0000-0001-6610-445X
Ryan Suttle http://orcid.org/0000-0002-8527-0098
Sanjay Arwade http://orcid.org/0000-0001-9971-828X
Brian Kane http://orcid.org/0000-0002-2814-0801

Data availability statement

All data and code generated for this study are publicly available in a GitHub repository online at https://github.com/narslab/inference-tree-risk.

References

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O. ... Farhan, L. (2021, March). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, *8*(1), 53. https://doi.org/10.1186/s40537-021-00444-8

Bäuerle, A., van Onzenoodt, C., & Ropinski, T. (2021). Net2vis – a visual grammar for automatically generating publication-tailored CNN architecture visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 1–1.

Brigato, L., & Iocchi, L. (2020, October). A close look at deep learning with small data. *arXiv:2003.12843 [cs, stat]* .

Chattopadhyay, A., Sarkar, A., Howlader, P., & Balasubramanian, V. N. (2018, March). Grad-CAM++: Improved visual explanations for deep convolutional networks. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Lake Tahoe, Nevada, USA (pp. 839–847).

Chollet, F. (2017, April). Xception: Deep learning with depthwise separable convolutions (No. arXiv:1610.02357). arXiv.

Denker, J., Gardner, W., Graf, H., Henderson, D., Howard, R., Hubbard, W. Guyon, I. (1988). Neural network recognizer for hand-written zip code digits. *Advances in Neural Information Processing Systems*, *1*, 323–331.

Donti, P. L., & Kolter, J. Z. (2021). Machine learning for sustainable energy systems. *Annual Review of Environment and Resources*, *46*(1), 719–747. https://doi.org/10.1146/annurev-environ-020220-061831

dos Santos, A. A., Marcato Junior, J., Araújo, M. S., DiMartini, D. R., Tetila, E. C., Siqueira, H. L. Gonc¸alves, W. N. (2019, January). Assessment of

CNN-Based methods for individual tree detection on images captured by RGB cameras attached to UAVs. *Sensors*, *19*(16), 3595. https://doi.org/10.3390/s19163595

Egli, S., & Höpke, M. (2020, January). CNN-Based tree species classification using high resolution RGB image data from automated UAV observations. *Remote Sensing*, *12*(23), 3892. https://doi.org/10.3390/rs12233892

Elliott, S. N., Shields, A. J. B., Klaehn, E. M., & Tien, I. (2022, January). Identifying critical infrastructure in imagery data using explainable convolutional neural networks. *Remote Sensing*, *14*(21), 5331. https://doi.org/10.3390/rs14215331

Fricker, G. A., Ventura, J. D., Wolf, J. A., North, M. P., Davis, F. W., & Franklin, J. (2019, January). A convolutional neural network classifier identifies tree species in mixed-conifer forest from hyperspectral imagery. *Remote Sensing*, *11*(19), 2326. https://doi.org/10.3390/rs11192326

Fukushima, K. (1975, September). Cognitron: A self-organizing multilayered neural network. *Biological Cybernetics*, *20*(3–4), 121–136. https://doi.org/10.1007/BF00342633

Fukushima, K. (1980, April). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, *36*(4), 193–202. https://doi.org/10.1007/BF00344251

Fukushima, K. (1988, January). Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural Networks*, *1*(2), 119–130. https://doi.org/10.1016/0893-6080(88)90014-7

Gazzea, M., Pacevicius, M., Dammann, D. O., Sapronova, A., Lunde, T. M., & Arghandeh, R. (2021). Automated power lines vegetation monitoring using HighResolution satellite imagery. *IEEE Transactions on Power Delivery*, *37*(1), 308–316. https://doi.org/10.1109/TPWRD.2021.3059307

Giebel, H. (1971). Feature extraction and recognition of handwritten characters by homogeneous layers. In O.-J. Grüsser & R. Klinke (Eds.), *Zeichenerkennung durch biologische und technische Systeme/Pattern Recognition in Biological and Technical Systems* (pp. 162–169). Springer. https://doi.org/10.1007/978-3-642-65175-5_15

Glorot, X., Bordes, A., & Bengio, Y. (2011, June). Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, Fort Lauderdale, Florida, USA (pp. 315–323). JMLR Workshop and Conference Proceedings.

Goodfellow, J. W. (2020). Best management practices - utility tree risk assessment (Tech. Rep. No. P1321). International Society of Arboriculture.

Graziano, M., Gunther, P., Gallaher, A., Carstensen, F. V., & Becker, B. (2020, March). The wider regional benefits of power grids improved resilience through tree-trimming operations evidences from Connecticut, USA. *Energy Policy*, *138*, 111293. https://doi.org/10.1016/j.enpol.2020.111293

Guggenmoos, S. (2003). Effects of tree mortality on power line security. *Journal of Arboriculture*, *29*(4), 181–196. https://doi.org/10.48044/jauf.2003.022

Guggenmoos, S. (2011). Tree-related electric outages due to wind loading. *Arboriculture & Urban Forestry*, *37*(4), 147–151. https://doi.org/10.48044/jauf.2011.019

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B. Chen, T. (2018, May). Recent advances in convolutional neural networks. *Pattern Recognition*, *77*, 354–377. https://doi.org/10.1016/j.patcog.2017.10.013

He, K., Zhang, X., Ren, S., & Sun, J. (2015, December). Deep residual learning for image recognition. *arXiv:1512.03385 [cs]* .

Hubel, D. H., & Wiesel, T. N. (1959, October). Receptive fields of single neurones in the cat's striate cortex. *The Journal of Physiology*, *148*(3), 574–591. https://doi.org/10.1113/jphysiol.1959.sp006308

Hubel, D. H., & Wiesel, T. N. (1962, January). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of Physiology*, *160*(1), 106–154.2. https://doi.org/10.1113/jphysiol.1962.sp006837

Hubel, D. H., & Wiesel, T. N. (1965, March). Receptive fields and functional architecture in two nonstriate visual areas (18 and 19) of the cat. *Journal of Neurophysiology*, *28*(2), 229–289. https://doi.org/10.1152/jn.1965.28.2.229

Hubel, D. H., & Wiesel, T. N. (1968). Receptive fields and functional architecture of monkey striate cortex. *The Journal of Physiology*, *195*(1), 215–243. https://doi.org/10.1113/jphysiol.1968.sp008455

Ioffe, S., & Szegedy, C. (2015, March). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv:1502.03167 [cs]* .

Jiao, P., & Alavi, A. H. (2020, May). Artificial intelligence in seismology: Advent, performance and future trends. *Geoscience Frontiers*, *11*(3), 739–744. https://doi.org/10.1016/j.gsf.2019.10.004

Kabrisky, M. (1966). *A proposed model for visual information processing in the human brain*. University of Illinois Press.

Kingma, D. P., & Ba, J. (2017, January). Adam: A method for stochastic optimization. *arXiv:1412.6980 [cs]*.

Koeser, A. K., McLean, D. C., Hasing, G., & Allison, R. B. Frequency, severity, and detectability of internal trunk decay of street tree Quercus spp. (2016). *Arboriculture & Urban Forestry*, *42*(4), 217–226. in Tampa, Florida, U.S. https://doi.org/10.48044/jauf.2016.020

Koeser, A. K., Thomas Smiley, E., Hauer, R. J., Kane, B., Klein, R. W., Landry, S. M., & Sherwood, M. (2020, June). Can professionals gauge likelihood of failure? – Insights from tropical storm Matthew. *Urban Forestry & Urban Greening*, *52*, 126701. https://doi.org/10.1016/j.ufug.2020.126701

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, *25*, 1097–1105.

LeCun, Y., Boser, B. E., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. E., & Jackel, L. D. (1989). Handwritten digit recognition with a back-propagation network. *Advances in Neural Information Processing Systems*, *2*, 9.

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998, November). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324. https://doi.org/10.1109/5.726791

Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., & Talwalkar, A. (2018). Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, *18*(185), 1–52.

Matikainen, L., Lehtomäki, M., Ahokas, E., Hyyppä, J., Karjalainen, M., Jaakkola, A. Heinonen, T. (2016,

September). Remote sensing methods for power line corridor surveys. *ISPRS Journal of Photogrammetry and Remote Sensing*, *119*, 10–31. https://doi.org/10.1016/j.isprsjprs.2016.04.011

McMillan, L., & Varga, L. (2022, November). A review of the use of artificial intelligence methods in infrastructure systems. *Engineering Applications of Artificial Intelligence*, *116*, 105472. https://doi.org/10.1016/j.engappai.2022.105472

Nateghi, R., Guikema, S., & Quiring, S. M. (2014). Power outage estimation for tropical cyclones: Improved accuracy with simpler models. *Risk Analysis*, *34*(6), 1069–1078. https://doi.org/10.1111/risa.12131

Nguyen, V. N., Jenssen, R., & Roverso, D. (2018, July). Automatic autonomous vision based power line inspection: A review of current status and the potential role of deep learning. *International Journal of Electrical Power & Energy Systems*, *99*, 107–120. https://doi.org/10.1016/j.ijepes.2017.12.016

Oliveira, A. A., Buckeridge, M. S., & Hirata, R. (2022, May). Detecting tree and wire entanglements with deep learning. *Trees*, *37*(1), 147–159. https://doi.org/10.1007/s00468-022-02305-0

Parent, J. R., Meyer, T. H., Volin, J. C., Fahey, R. T., & Witharana, C. (2019, July). An analysis of enhanced tree trimming effectiveness on reducing power outages. *Journal of Environmental Management*, *241*, 397–406. https://doi.org/10.1016/j.jenvman.2019.04.027

Rooney, C. J., Ryan, H. D., Bloniarz, D. V., & Kane, B. (2005). THE reliability of a windshield survey to locate hazards in roadside trees. *Journal of Arboriculture*, *31*(2), 89–94. https://doi.org/10.48044/jauf.2005.011

Rosenblatt, F. (1962). *Principles of neurodynamics; perceptrons and the theory of brain mechanisms*. Spartan Books.

Salehi, H., & Burgueño, R. (2018, September). Emerging artificial intelligence methods in structural engineering. *Engineering Structures*, *171*, 170–189. https://doi.org/10.1016/j.engstruct.2018.05.084

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2020, February). Grad-CAM: Visual explanations from deep networks via gradient based localization. *International Journal of Computer Vision*, *128* (2), 336–359. https://doi.org/10.1007/s11263-019-01228-7

Shorten, C., & Khoshgoftaar, T. M. (2019, July). A survey on image data augmentation for deep learning. *Journal of Big Data*, *6*(1), 60. https://doi.org/10.1186/s40537-019-0197-0

Simonyan, K., & Zisserman, A. (2015, April). Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556 [cs]*.

Simpson, P., & Van Bossuyt, R. (1996). TREE-CAUSED ELECTRIC OUTAGES. *Journal of Arboriculture*, *22*(3), 117–121. https://doi.org/10.48044/jauf.1996.018

Smiley, E. T., Matheny, N., & Lilly, S. (2017). Best management practices - tree risk assessment, second edition (Tech. Rep. No. P1542). International Society of Arboriculture.

Spencer, B. F., Hoskere, V., & Narazaki, Y. (2019, April). Advances in computer vision-based civil infrastructure inspection and monitoring. *Engineering*, *5*(2), 199–222. https://doi.org/10.1016/j.eng.2018.11.030

Su, Z., Chow, J. K., Tan, P. S., Wu, J., Ho, Y. K., & Wang, Y.-H. (2020, October). Deep convolutional neural network–based pixel-wise landslide inventory mapping. *Landslides*, *18*(4), 1421–1443. https://doi.org/10.1007/s10346-020-01557-6

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2016, August). Inception- v4, Inception-ResNet and the impact of residual connections on learning. *arXiv:1602.07261 [cs]* .

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D. Rabinovich, A. (2014, September). Going deeper with convolutions. *arXiv:1409.4842 [cs]* .

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2015, December). Rethinking the inception architecture for computer vision. *arXiv:1512.00567* [cs] .

Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., and Tang, X. (2017, July). Residual attention network for image classification. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, Hawaii, USA (pp. 6450–6458).

Wang, N., Zhao, X., Wang, L., & Zou, Z. (2019, September). Novel system for rapid investigation and damage detection in cultural heritage conservation based on deep learning. *Journal of Infrastructure Systems*, *25*(3), 04019020. https://doi.org/10.1061/(ASCE)IS.1943-555X.0000499

Watanabe, J.-I., Ren, S., Zhao, Y., & Yamamoto, T. (2018, May). Power line-tree conflict detection and 3D mapping using aerial images taken from UAV. In *2018 SPIE Defense + Security*, Orlando, Florida, USA (Vol. 10643). SPIE Press.

Wismer, S. (2018, May). *Targeted Tree Trimming Offers Reliability Benefits*. https://www.tdworld.com/vegetation-management/article/20971285/targeted- tree-trimming-offers-reliability-benefits.

Wong, S. C., Gatt, A., Stamatescu, V., & McDonnell, M. D. (2016, November). Understanding data augmentation for classification: When to warp? *arXiv:1609.08764 [cs]* .

Woo, S., Park, J., Lee, J.-Y., & Kweon, I. S. (2018, July). CBAM: Convolutional block attention module. (No. arXiv:1807.06521). arXiv.

Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2017, April). Aggregated residual transformations for deep neural networks. (No. arXiv:1611.05431). arXiv.

Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In D. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars (Eds.), *Computer Vision – ECCV 2014*, (pp. 818–833). Springer International Publishing.

Zhang, Y., Yuan, X., Li, W., & Chen, S. (2017, August). Automatic power line inspection using UAV Images. *Remote Sensing*, *9*(8), 824. https://doi.org/10.3390/rs9080824

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2016, June). Learning deep features for discriminative localization. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, Nevada, USA (pp. 2921–2929).