

Western University

Scholarship@Western

---

Civil and Environmental Engineering  
Publications

Civil and Environmental Engineering  
Department

---

2-13-2024

## Automated large-scale tornado treefall detection and directional analysis using machine learning

Daniel G. Butt

Aaron L. Jaffe

Connell S. Miller

Gregory A. Kopp

David Sills

Follow this and additional works at: <https://ir.lib.uwo.ca/civilpub>



Part of the [Civil and Environmental Engineering Commons](#)

---

# Automated Large-Scale Tornado Treefall Detection and Directional Analysis Using Machine Learning

DANIEL G. BUTT,<sup>a</sup> AARON L. JAFFE,<sup>a</sup> CONNELL S. MILLER<sup>ORCID</sup>,<sup>a</sup> GREGORY A. KOPP,<sup>a</sup> AND DAVID M. L. SILLS<sup>a</sup>

<sup>a</sup> *Northern Tornadoes Project, Faculty of Engineering, Western University, London, Ontario, Canada*

(Manuscript received 27 July 2023, in final form 9 November 2023, accepted 10 January 2024)

**ABSTRACT:** In many regions of the world, tornadoes travel through forested areas with low population densities, making downed trees the only observable damage indicator. Current methods in the EF scale for analyzing tree damage may not reflect the true intensity of some tornadoes. However, new methods have been developed that use the number of trees downed or treefall directions from high-resolution aerial imagery to provide an estimate of maximum wind speed. Treefall Identification and Direction Analysis (TrIDA) maps are used to identify areas of treefall damage and treefall directions along the damage path. Currently, TrIDA maps are generated manually, but this is labor-intensive, often taking several days or weeks. To solve this, this paper describes a machine learning- and image-processing-based model that automatically extracts fallen trees from large-scale aerial imagery, assesses their fall directions, and produces an area-averaged treefall vector map with minimal initial human interaction. The automated model achieves a median tree direction difference of 13.3° when compared to the manual tree directions from the Alonsa, Manitoba, tornado, demonstrating the viability of the automated model compared to manual assessment. Overall, the automated production of treefall vector maps from large-scale aerial imagery significantly speeds up and reduces the labor required to create a Treefall Identification and Direction Analysis map from a matter of days or weeks to a matter of hours.

**SIGNIFICANCE STATEMENT:** The automation of treefall detection and direction is significant to the analyses of tornado paths and intensities. Previously, it would have taken a researcher multiple days to weeks to manually count and assess the directions of fallen trees in large-scale aerial photography of tornado damage. Through automation, analysis takes a matter of hours, with minimal initial human interaction. Tornado researchers will be able to use this automated process to help analyze and assess tornadoes and their enhanced Fujita-scale rating around the world.

**KEYWORDS:** Tornadoes; Aircraft observations; Unpiloted aerial systems; Automated systems; Deep learning; Neural networks

## 1. Introduction

The enhanced Fujita (EF) scale is implemented in several countries such as Canada, the United States, China, and Japan to assess the severity of tornadoes. Within this scale, various damage indicators (DIs) are evaluated, each having corresponding degrees of damage (DOD) along with their associated wind speeds. The spectrum of DODs typically spans from the point of minimal discernible damage to the complete devastation of the DI. (McDonald and Mehta 2006). The overall EF-scale rating for the damage is then assigned based on the maximum wind speed across all observed damage indicators (Mehta 2013).

In many regions of the world, tornadoes travel through forested areas with low population densities, meaning that downed trees are the only observable damage indicator. For example, Fig. 1 shows a map of Canadian tornadoes (over land) recorded by the Northern Tornadoes Project (NTP) from 2017 to 2022 that occurred in areas where the population density is less than 1 person km<sup>-2</sup> (Government of Canada 2020). Of the 510 tornadoes over land recorded by NTP, 259 of these (50.8%) occurred in these areas where the population density is less than 1 person km<sup>-2</sup>, with 81% of these having

forests along some portion of the track (Karra et al. 2021). Figure 2 shows an unmanned aerial vehicle (UAV) photo depicting significant treefall from a Canadian tornado.

There are international variations for how the EF scale handles damage to trees. In the United States, for example, there are two tree DIs: one for softwood and the other for hardwood. The Canadian version of the EF scale (Sills et al. 2014) has a single tree DI with most DODs related to the percentage of trees snapped or uprooted along the path of damage. The NTP developed a method for accurately and consistently estimating those percentages called the “scalable box method” (Sills et al. 2020), where a box (with size related to tornado width) is drawn around the area of worst damage and the percentage of downed trees in that area is used to provide an estimate of the maximum wind speed. The maximum EF-scale rating a tornado can be assigned via this DI is EF3, which may not reflect the true intensity of some tornadoes. Additionally, many of these tornadoes occur on the shallow soil of the Canadian Shield, where downed trees can only be given a maximum rating of EF2.

Significant damaging wind events that occur in remote areas or cause extensive tree damage are often analyzed with the help of high-resolution aerial imagery (Sills et al. 2020). Aerial imagery consists of aircraft and UAV imagery, one or both of which may be collected for an event depending on the severity and remoteness of the damage. Typical high-resolution

Corresponding author: Connell S. Miller, connell.miller@uwo.ca

DOI: 10.1175/AIES-D-23-0062.1 e230062

© 2024 American Meteorological Society. This published article is licensed under the terms of the default AMS reuse license. For information regarding reuse of this content and general copyright information, consult the AMS Copyright Policy ([www.ametsoc.org/PUBSReuseLicenses](http://www.ametsoc.org/PUBSReuseLicenses)).

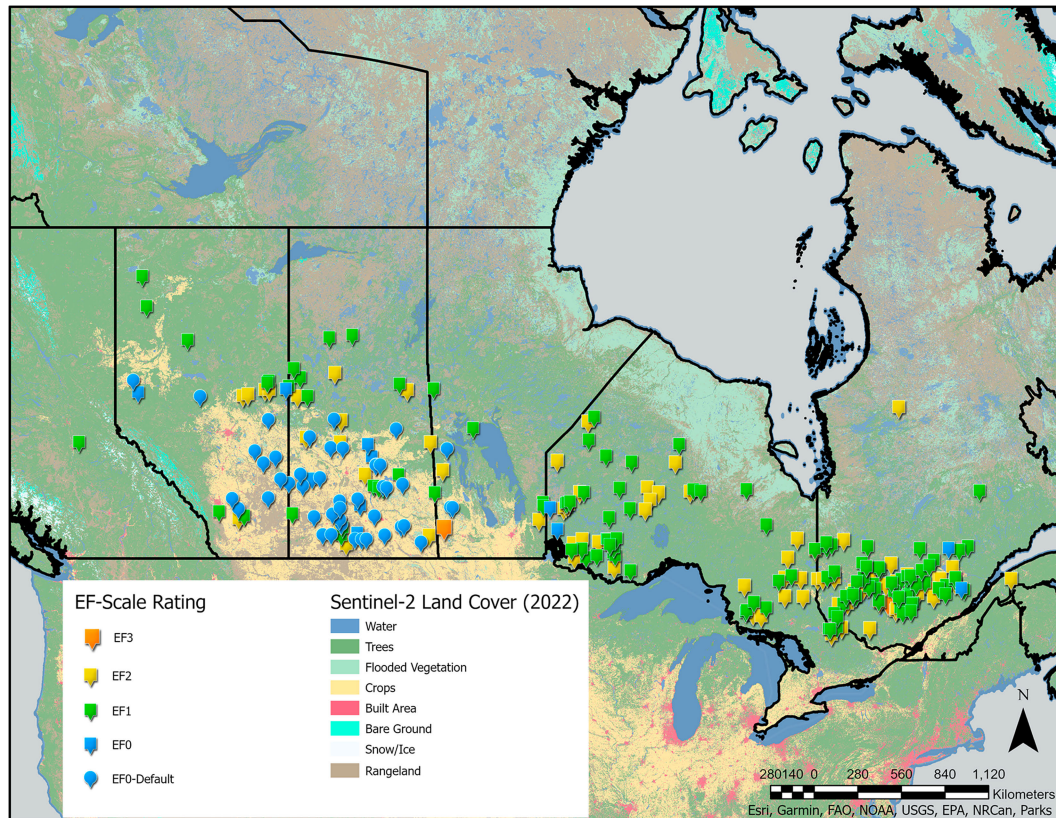


FIG. 1. Map depicting Canadian tornadoes (over land) recorded by the Northern Tornadoes Project from 2017 to 2022 in areas where the population density is less than 1 person  $\text{km}^{-2}$  as well as land cover recorded by Sentinel-2 data.

aircraft imagery used by research groups has a resolution of up to 5 cm per pixel and is captured weeks or months after an event. UAV imagery typically has a minimum resolution of 2.5 cm per pixel and is captured days or weeks after an event. Regardless of the aerial imagery collection method, these images can be stitched together to produce a high-resolution, georeferenced aerial map, also known as an orthomosaic map. These orthomosaics may cover an area of 1–100  $\text{km}^2$ , or more, and typically contain all the significant damage from an event in the case of aircraft imagery, or sometimes just a notable area of damage along the damage path in the case of UAV imagery. Importantly, the resolution of aircraft imagery is detailed enough to detect individual trees, as well as the direction in which they fell. The enhanced resolution of UAV imagery adds the capability to determine tree species as well as soil type and condition. Table 1 lists the specific details of the aerial imagery used in this study. The aerial imagery presented in this paper was taken on clear days during various hours of bright sunlight, which is critical for being able to clearly see the downed trees.

All aerial surveys of severe wind damage conducted by researchers result in some level of damage analysis. Typically, at minimum, damage from the wind event is manually outlined, the centerline, pathlength, and maximum width are determined, and a wind speed and associated EF rating are assigned. Since 2020, for particularly significant or

complicated events, especially those involving a mix of tornado and downburst damage, the NTP generates a Treefall Identification and Direction Analysis (TriDA) map. These analyses are inspired by similar work done by T. T. Fujita in several of his papers (e.g., Fujita and Wakimoto 1981; Fujita 1989). Generating a TriDA map consists of identifying the areas of treefall damage and the general treefall directions along the damage path. First, all areas of fresh treefall are enclosed by polygons to highlight the damage path. Then, the average treefall directions of groups of trees are noted, spaced as needed to obtain a good understanding of the treefall patterns in the damage. Finally, when applicable, these treefall directions can be used to distinguish between tornadic and downburst damage. Generally speaking, tornadic damage is convergent and closer to the damage centerline, whereas related downburst damage is usually divergent and off the main path of the tornado. Occasionally, the entire area of damage is divergent and caused by one or multiple downbursts, which may not be known until after the analysis is performed. TriDA maps are useful for separating out potential downburst damage from related tornado paths while also providing some insight into the character of the event and a valuable visual representation of the wind patterns.

Currently, TriDA maps are generated manually. The tagging of areas of tree damage and general treefall directions is



FIG. 2. A UAV photo of significant treefall, including a clear swirl pattern (foreground), from the 2018 EF2 Saint-Julien tornado in Quebec.

time consuming, often taking several days or weeks, depending on the size of the event. There is also reason to use these analyses on more than just complicated events with a mix of tornado and downburst damage. As mentioned earlier, the EF scale and scalable box method do not usually allow for the rating of Canadian tornadoes above EF2 when only forest damage is present, as most of them occur on the shallow soil of the Canadian Shield. However, alternative methods have been developed that use treefall directions in significant tornadoes to provide a maximum wind speed for these events. Some notable examples include [Karstens et al. \(2013\)](#), [Godfrey and Peterson \(2017\)](#), and [Rhee and Lombardo \(2018\)](#). These methods present

the possibility of assessing EF3 or higher damage from tornadoes in forested areas at the cost of requiring significant treefall data for their analyses. Godfrey and Peterson’s method requires the identification of every downed tree, with Karsten’s or Rhee and Lombardo’s method requiring area-averaged treefall directions. Compared to other analysis methods, such as the scalable box method and ground surveys, wind speeds determined by these treefall methods tend to be higher. This development has increased the need and utility of TrIDA maps and the desire to make these analyses more time efficient. To address these needs, an automated process to assist with the generation of TrIDA maps is required.

In 2017, a method for coarse-to-fine extraction of downed trees from UAVs was proposed by [Duan et al. \(2017\)](#). Their extraction method first utilizes a random forest machine learning model ([Breiman 2001](#)) to extract a rough mask of the trees, followed by image processing techniques that leverage the linear shape of trees to refine the mask. Once a refined mask is produced, the Hough transform ([Hough 1962](#)) is implemented to fit lines to the tree stems. However, their dataset was limited to a single event in northeastern Hainan, China, and taken from a UAV camera at a relatively high altitude of 500 m, producing lower-resolution (10 cm per pixel) imagery.

TABLE 1. List of the specifications of the aerial imagery used in this study.

Aerial imagery specifications	
Resolution	20000 × 20000 pixel per image
Scale	5 cm × 5 cm per pixel
Area	1 km <sup>2</sup> per image
Color channels	Red, green, blue (24-bit RGB)
Image format	.tif, no compression

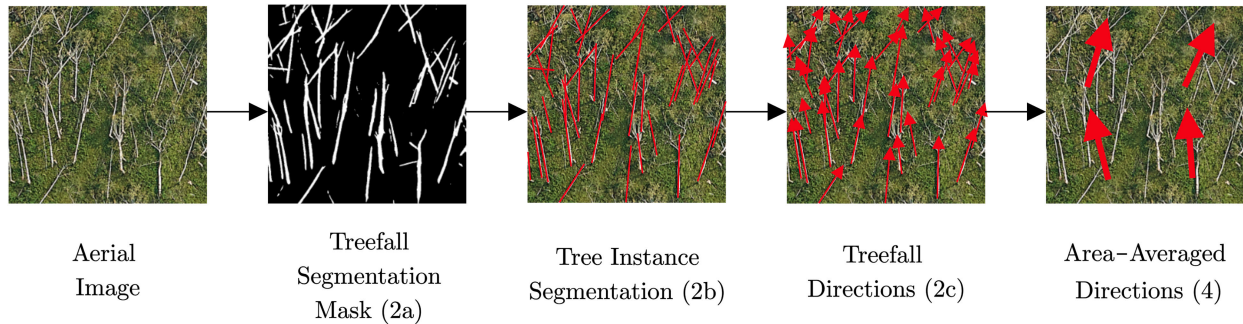


FIG. 3. High-level flowchart for the automated treefall model. The number and letter next to each image indicates the relevant section of the paper where each part is covered.

As a result, many of their techniques would require significant manual adjustment if applied to a broader range of tornadic events with higher concentrations of trees or different species of trees with more pronounced branches that are resolved with higher-resolution imagery.

In early 2021, a semiautomated method for identification of downed trees from 5-cm aerial photography was proposed by Rhee et al. (2021). Their method utilizes many image processing techniques including image filtering to extract tree stems and leaves. After extraction, the Hough transform is used to fit lines to the stems, and the position of leaves relative to these lines is used to extrapolate the area-averaged tree directions for a given area. Although effective at extracting trees, the process is only semiautomated, requiring significant adjustments and human interaction to achieve accurate results. Moreover, their direction-extraction method uses leaf positions, which does not work in areas with a high concentration of downed trees, areas where the colors of the leaves are similar to the background colors, or with trees that do not have leaves.

Most recently, the detection of tree stems from UAV orthomosaics using U-Net convolutional neural networks was demonstrated by Reder et al. (2022). They created various datasets augmented from 454 trees downed in a severe storm northeast of Berlin, Germany. These datasets are then used to train a U-Net model to perform semantic segmentation of downed tree stems (Ronneberger et al. 2015). Their results are automated and are more effective at extracting tree stems than the coarse-to-fine method proposed by Duan et al. (2017). However, their datasets are limited to a single event with a moderate density of trees, and no fitting of lines or extraction of direction is performed.

Given the above, a method for systematic wide-ranging treefall analysis in remote forested areas is needed. The objective of this paper is to describe a machine learning and image processing-based model that can automatically extract fallen trees from large-scale aerial imagery, assess their fall directions, and produce an area-averaged treefall vector map with minimal initial human interaction.

## 2. Methodology

The method developed to produce an automated treefall map model is described in this section and summarized in Fig. 3. First,

a treefall segmentation mask is produced similar to that of Reder et al. (2022). Next, instance segmentation is performed using the segmentation mask to extract individual trees, followed by assessing their fall directions. Finally, the treefall directions are sorted into a chosen grid, and the area-averaged directions of trees in each grid square are used to create a treefall vector map.

### a. Treefall semantic segmentation model

To identify each fallen tree contained in the aerial images collected from a tornado swath through a forest, it is beneficial to first extract a binary segmentation mask containing just the pixels of the fallen tree stems. This mask highlights only the pixels that are part of tree stems, removing everything else. Producing a segmentation mask removes noise and simplifies the image, making further image processing significantly more effective, as demonstrated by Duan et al. (2017) and Rhee et al. (2021). To create a binary segmentation mask of the tree-stem pixels, a U-Net architecture is used similar to Reder et al. (2022).

The U-Net deep-learning architecture is a type of fully convolutional neural network originally designed for the purposes of biomedical image segmentation (Ronneberger et al. 2015). Semantic segmentation is the process of labeling each pixel in an image as belonging to a specific equivalence class. More specifically, U-Net performs semantic segmentation by first using a series of convolutional layers in conjunction with downsampling using max-pooling (much like a typical convolutional neural network), referred to as an encoder or the backbone of the network. Then, a series of repeated convolutional layers followed by upsampling are combined with the extracted features from the encoder to ultimately make pixel-wise predictions. The U-Net architectures can be adjusted to take any fixed-sized images as an input, but generally smaller sizes ranging between  $128 \times 128$  and  $572 \times 572$  pixels are used due to the computational complexity with larger image sizes. Moreover, different encoders (backbones) including residual networks (ResNet; He et al. 2016) and Visual Geometry Group's (VGG) convolution neural networks (Simonyan and Zisserman 2014) can be substituted.

To construct a dataset to train the U-Net model, 10 images with an average size of  $200 \text{ m} \times 200 \text{ m}$  ( $4000 \times 4000$  pixels), are chosen from the seven tornadoes listed in Table 2. The

TABLE 2. List of tornadic events used for both the training and validation datasets of the treefall semantic segmentation model.

Event name	Location (province)	Date	Canadian EF-scale rating	No. of images used for training	No. of images used for validation	No. of images used for empirical tests
Alonsa	Manitoba	3 Aug 2018	EF4	0	1	1
Brooks Lake	Ontario	8 Jun 2020	EF2	2	1	0
Lac Flocon	Quebec	5 Sep 2018	EF2	2	0	0
Lac Gus	Quebec	5 Sep 2018	EF2	0	1	1
Lac Rouille	Quebec	5 Sep 2018	EF2	1	0	0
Lake Traverse	Ontario	15 Jun 2021	EF2	2	0	0
Mary Lake	Ontario	10 Jun 2020	EF2	0	0	1

fallen trees in these images are manually segmented to produce segmentation masks, as shown in Fig. 4. Only 10 images were utilized to reduce the required time for manual segmentation and quality control. Three images are chosen for validation, while the other seven are used for training, also shown in Table 2. The validation images are chosen from larger events with more imagery or because later evaluation of the model's performance would be done on the same events. For evaluation purposes, no images from the 2018 EF4 Alonsa, Manitoba, or the 2018 EF2 Lac Gus, Quebec, tornadoes are used for training. A size of  $256 \times 256$  pixels ( $12.8 \text{ m} \times 12.8 \text{ m}$ ) is chosen as the input image size for the U-Net model as it gives a good balance between real-world scale, computational complexity, and the size of the produced dataset. Each of the manually segmented images and their corresponding segmentation masks are then split into  $256 \times 256$  pixel images using grids with various offsets, followed by performing data augmentation techniques including rotating, flipping, and adjusting of hue. Data augmentation helps to enhance diversity and increase the overall dataset size leading to improvements in training accuracy as shown by Reder et al. (2022). The final dataset contains just over 100 000 and 35 000,  $256 \times 256$  pixel images and corresponding tree segmentation masks for training and validation, respectively.

The TensorFlow (Abadi et al. 2016) and Keras (Chollet et al. 2015) Python modules are used for building and training the U-Net models. During training, a variety of backbone

convolutional networks including ResNet-18/34/50 and VGG-16/19 (the numbers denoting the number of convolutional layers) are used to achieve the best fit as well as compare performance between smaller and larger models. For a loss function, the Dice similarity coefficient, the harmonic mean of precision and recall (Dice 1945), and weighted binary cross-entropy (WBCE) are used, as shown by (Reder et al. 2022). These loss functions are chosen due to the unbalanced nature of the dataset having many more non-tree-stem pixels than tree-stem pixels. The ratio of tree-stem pixels to non-tree-stem pixels in the produced dataset is approximately 1:10. As a result, weights of 15:1 are used with WBCE in order to prioritize the extraction of tree-stem pixels over non-tree-stem pixels. This strong weighting likely increases the number of falsely identified tree-stem pixels. However, this is preferable to the alternative possibility of missing trees altogether. The adaptive momentum (ADAM) optimizer (Kingma and Ba 2014), along with a learning rate of 0.01 and batch size of 16, are used to optimize model weights, with all models being trained until validation loss converges.

Once training is complete, a segmentation mask of a large-scale aerial image is produced by splitting the image into  $256 \times 256$  pixel sections, with each section being processed by the U-Net model. After processing, each individual section is stitched back together to reassemble the sections into a complete segmentation mask of the original image. This process can be seen in Fig. 5.

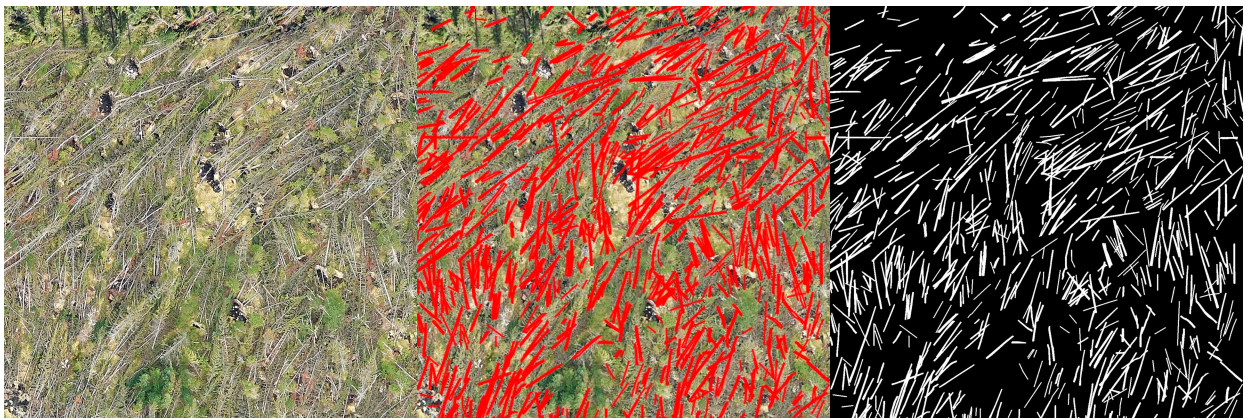


FIG. 4. A  $76.8 \text{ m} \times 76.8 \text{ m}$  section of the aerial imagery from (left) the 2018 EF2 Lac Gus tornado in Quebec, (center) the manual tree segmentation, and (right) the manually produced binary tree segmentation mask.

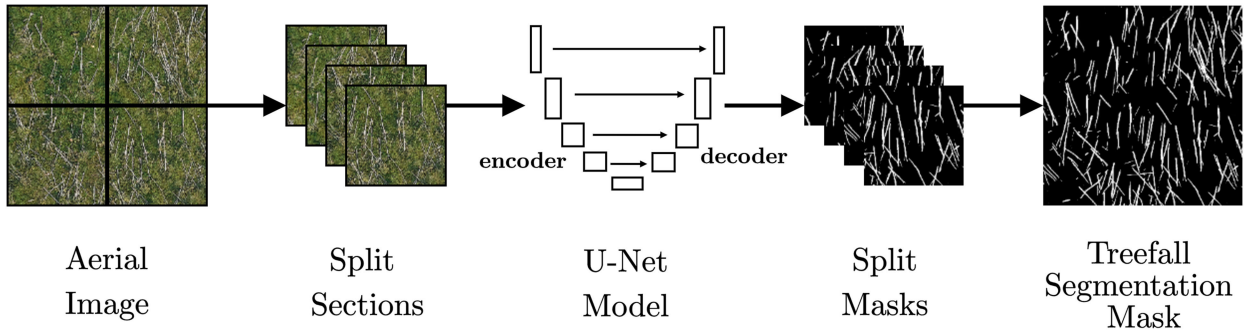


FIG. 5. A flowchart demonstrating the processes of producing a treefall segmentation mask of a large-scale aerial image using a U-Net deep learning model.

*b. Tree instance segmentation algorithm*

Following semantic segmentation, instance segmentation is performed in order to separate and identify individual trees so that their directions can be assessed. For assessment of

direction, it is beneficial to perform instance segmentation by fitting a line to the stem of each tree, referred to as “tree tagging.” In the works of [Duan et al. \(2017\)](#) and [Rhee et al. \(2021\)](#), the Hough transform is utilized for instance segmentation. However,

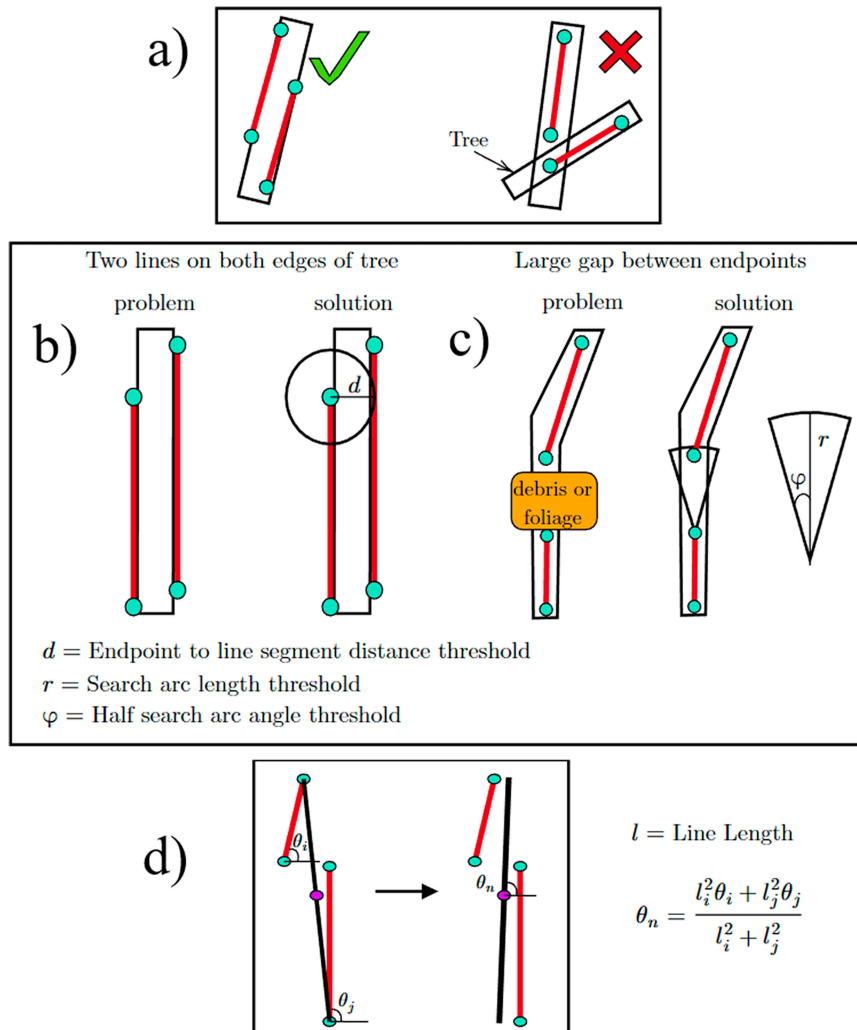


FIG. 6. A diagram that demonstrates the key ideas utilized in the line-joining algorithm. (a) The angle difference criterion. (b) The endpoint to line segment distance criterion. (c) The search arc criterion, and (d) a demonstration of the line-joining process.

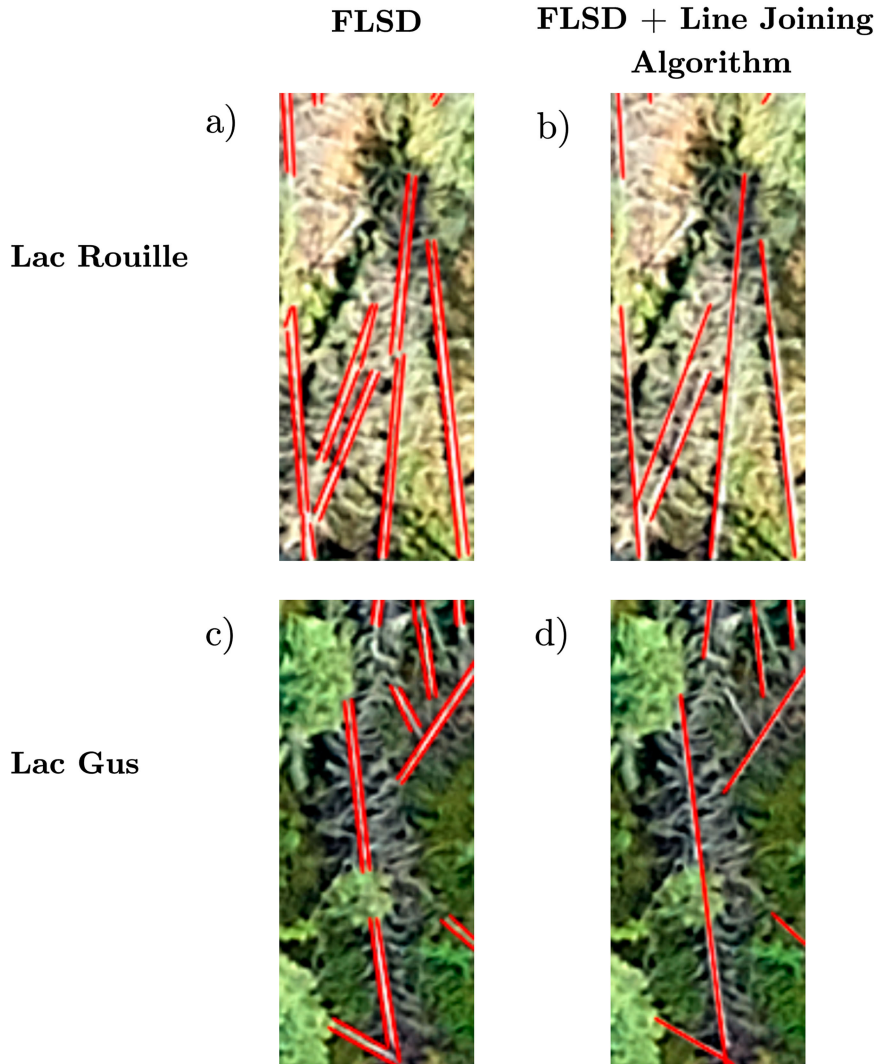


FIG. 7. (a),(c) Examples of the output of the fast line segment detection algorithm (van Gioi et al. 2010), from the 2018 EF2 Lac Rouille/Gus tornadoes in Quebec. (b),(d) The results from the line-joining algorithm in (a) and (c), respectively.

initial testing showed that the Hough transform struggles to accurately segment trees in high-density areas where many trees are overlapping. As a result, edge-based line segment detection algorithms are utilized instead.

After the production of the tree segmentation mask, the mask is preprocessed using the Guo and Hall (1992) thinning algorithm, followed by applying a  $3 \times 3$  dilation and mean blur convolution. This preprocessing stage is done to ensure the trees in the segmentation mask are of uniform thickness and to facilitate ideal conditions for extraction using edge-based line segment detectors. Once preprocessed, the fast line segment detection algorithm (von Gioi et al. 2010) is used to fit lines to the edges of the segmented tree-stem pixels. However, edge-based line segment detectors fit lines to each edge of the tree stems, so to join together the edges, a line-joining algorithm is developed.

The line-joining algorithm first uses spatial hashing to sort the found edge lines into grid squares of the same  $256 \times 256$  pixel ( $12.8 \text{ m} \times 12.8 \text{ m}$ ) size used with the production of the treefall segmentation mask. This is done to improve computational performance since lines only need to be compared with other lines in a  $3 \times 3$  grid around the same grid square, given that the height of tree stems in the tornado events used are shorter than  $3 \times 12.8 \text{ m} = 38.4 \text{ m}$ . Each line is then compared to all other lines in or around its corresponding grid square and, if a set of criteria is met, the lines are joined into a single line before being added back to the same grid square. The first criterion is that the lines should be close to parallel with each other, such that the angle difference between the lines is less than an angle difference threshold  $\Delta\theta$ . This criterion is based on the fact that most tree stems are linear in shape and lines that are not parallel to each other are likely different trees as described in Fig. 6a.



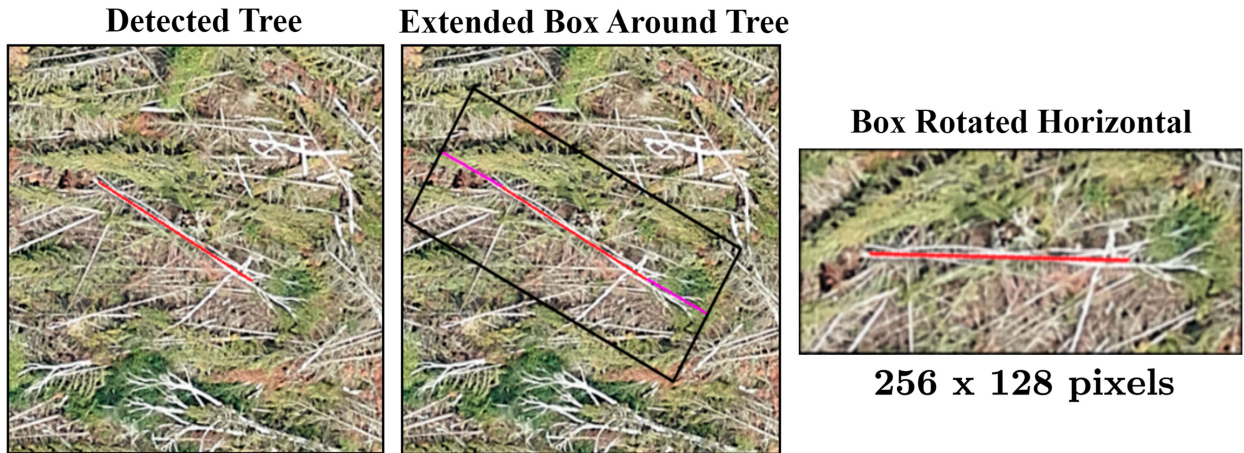


FIG. 8. A diagram that demonstrates the process of converting a tagged tree identified by the tree instance segmentation algorithm into a horizontally aligned image.

Next, the distance from an endpoint of one line, to the opposite line segment must be less than a distance threshold  $d$ . This criterion is utilized to join together lines that are on either edge of a tree stem, as shown in Fig. 6b. Then, search arcs of radius  $r$  and angle  $2\varphi$  are extended out from the endpoints of one line, to see if one of the opposite line's endpoints fall within the search arc. This criterion is utilized to join lines that are separated by sections of debris or foliage blocking the middle of a tree stem from being picked up by the semantic segmentation model as demonstrated in Fig. 6c. In order for a pair of lines to be joined, they must meet the minimum angle difference threshold and either the minimum endpoint distance threshold or search arc criteria.

When joining two lines together, a line between the furthest apart pair of endpoints between the first and second line is considered. Next, a weighted average of the angles of the first and second lines is calculated, weighting the angles based on the squared length of their lines. Then, the line between the farthest endpoints is rotated to match this angle as shown in Fig. 6d. The described weighted average skews the angle more toward lines of longer length. This is used since lines of longer length tend to be less prone to slight curves in a tree stem, which can be picked up by edge-based line segmented detectors. Last, after all lines have been considered and joined as necessary, any lines found to have a length less than  $m$  are removed, being considered as false positives, most likely due

to branches or other linearly shaped debris. Example results of the line-joining algorithm can be seen in Fig. 7.

The value of  $d$  is set to  $3\sqrt{2}$ , based on the uniform width of trees following the preprocessing of the tree segmentation mask and  $3 \times 3$  dilation/blur convolutions. Next, the value of  $m$  is set at 1.5 m as trees of length less than 1.5 m are uncommon throughout the various tornadic events tested, being likely false positives. Then, the values of  $\Delta\theta$ ,  $\varphi$ , and  $r$  are experimentally fit to  $9^\circ$ ,  $1.8^\circ$ , and 1.5 m, respectively, by finding the average of optimal values over the various tornado events tested.

### c. Tree direction model

Once instance segmentation has been performed to find and fit lines to detected trees in the imagery, each fallen tree's direction must be assessed to convert the fitted line into a vector. Mathematically, the direction or the angle of each fallen tree is a continuous value ranging over  $[0^\circ-360^\circ)$  before looping back to  $0^\circ$  for values larger than or equal to  $360^\circ$ . Although machine learning algorithms can certainly be utilized to directly predict continuous values, training a machine learning model to understand the similarities between  $0^\circ$  and  $360^\circ$  poses an additional challenge. However, a line segment has already been fit to each detected tree; thus, only the direction of the line needs to be assessed. Given this simplification,

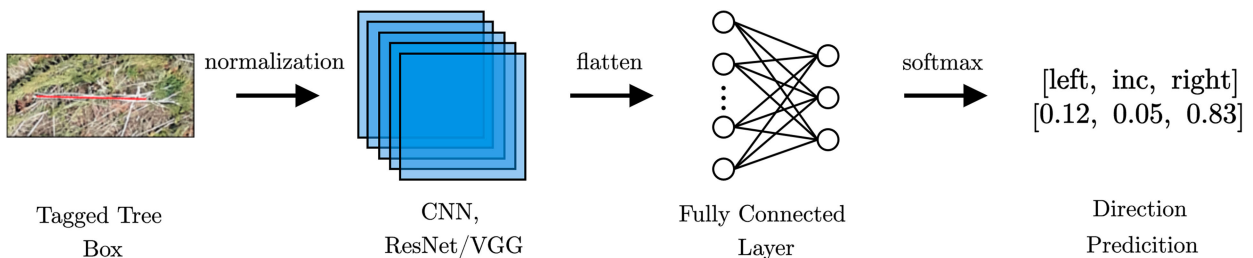


FIG. 9. A flowchart that demonstrates the use of convolutional neural networks along with fully connected layers for classification of horizontally aligned tree box images as either left, right, or inconclusive (inc).

TABLE 3. Training results for U-Net tree segmentation model. The bold font indicates the best performance.

Model architecture	Loss function	Validation accuracy	Validation DICE (F1)	Validation WBCE
ResNet-50 U-Net	Dice	95.0%	94.9%	82.6%
ResNet-50 U-Net	Dice + WBCE	95.2%	94.3%	94.7%
ResNet-34 U-Net	Dice + WBCE	95.1%	94.8%	94.3%
ResNet-18 U-Net	Dice + WBCE	95.0%	94.1%	95.1%
VGG-19 U-Net	Dice	95.2%	95.0%	89.4%
VGG-19 U-Net	Dice + WBCE	95.5%	94.7%	95.3%
<b>VGG-16 U-Net</b>	<b>Dice + WBCE</b>	<b>95.4%</b>	<b>95.1%</b>	<b>94.2%</b>

a simpler model can be constructed to classify the direction as either pointing toward one of the line's endpoints or the other.

To perform this direction classification, a rectangular box of  $768 \times 384$  pixels ( $38.4 \text{ m} \times 19.2 \text{ m}$ ) is extended around the line on the image containing the detected tree. This size is chosen to ensure all trees fit within the boxes, while also containing relevant areas of the tree's surroundings. Next, this section of the image is cropped out, rotated such that it aligns with the horizontal axis, and resized to  $256 \times 128$  pixels, as shown in Fig. 8. Once horizontal, the classification of the direction of each line can be further simplified to either pointing toward the left or right. Additionally, for the purposes of filtering out false identification of branches/debris as tree stems or the inability to assess the direction, a third category of inconclusive is added. From here, convolutional neural networks such as ResNet and VGG along with the addition of fully connected layers can be utilized to make the required

classification of either left, right, or inconclusive, as shown in Fig. 9.

To train a convolutional neural network, the tree semantic segmentation model and instance segmentation algorithms are run on the same training and validation images used for the semantic segmentation model, along with an additional image from the 2020 EF2 Mary Lake, Ontario, and 2019 EF2 Lac des Iles, Saskatchewan, tornadoes being added to increase the diversity of the training dataset. This produces horizontal tree boxes for each detected tree as shown in Fig. 8. From here, 16 000 horizontal tree boxes are manually labeled as fallen to the left, right or inconclusive (which notes either a false positive or the inability to assess the fall direction manually). The same images from the Alonsa, Brooks Lake, and Lac Gus tornadoes are again separated for validation. Data augmentation techniques, including flipping and adjusting of hue are utilized, bringing the final dataset to just over 100 000 and 30 000 images for training and validation, respectively.

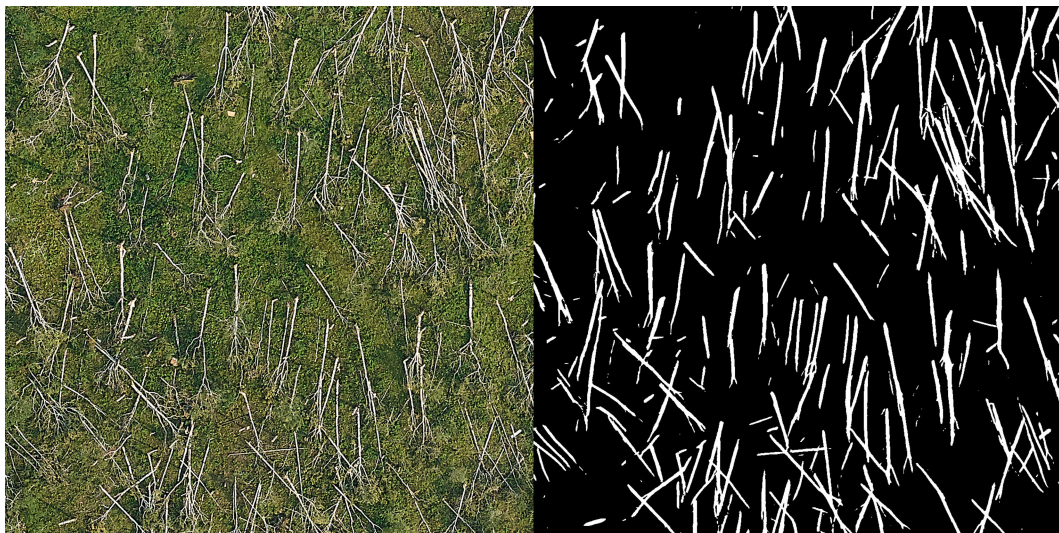


FIG. 10. (left) A  $64 \text{ m} \times 64 \text{ m}$  image of the Alonsa, Manitoba, tornado and (right) corresponding automatically extracted mask.



FIG. 11. (left) A  $64\text{ m} \times 64\text{ m}$  image of the Lac Gus, Quebec, tornado and (right) automatically extracted mask.

ResNet and VGG convolutional neural networks are selected with an input image size of  $256 \times 128$  pixels, downsampling the original  $768 \times 384$  pixel boxes to reduce the overall model size. The Keras Tuner Python module (Chollet et al. 2015), along with Bayesian optimization (Snoek et al. 2012), is utilized to find the optimal fully connected layer size, along with adjusting hyperparameters such as batch size and learning rate. The ADAM optimizer (Kingma and Ba 2014) is used along with the Dice score as a loss function. The Dice score is chosen due to the unbalanced nature of the training dataset at approximately a 45%/45%/10% split between left, right, and inconclusive, respectively. All tested models are trained until validation loss converges.

Once the convolutional neural network is trained, lines from each tagged tree can be automatically used to produce a horizontal tree box image and then have its direction classified.

Together the classified direction of left or right, along with the tagged tree line's original orientation, can be used to produce a fall direction vector for every tagged tree. In the case of a direction classification as inconclusive, the tagged tree is excluded from any further use.

### 3. Results

Table 3 shows the results of the U-Net tree segmentation models trained for a given combination of model architecture and loss function. The VGG-16 model using a Dice + weighted binary cross-entropy loss function performed the best with a validation Dice score of 95.1%. Due to the unbalanced nature of the dataset, 1:10 for tree-stem pixels to non-tree-stem pixels, the Dice score is considered a better indicator of performance than accuracy. However, all models performed quite similarly,



FIG. 12. (left) A  $64\text{ m} \times 64\text{ m}$  image of the Mary Lake, Ontario, tornado and (right) automatically extracted mask.



FIG. 13. (left) Preprocessed treefall segmentation mask, (center) extracted edge lines using fast line segment detector (van Goi et al. 2010), and (right) joined lines using the constructed line-joining algorithm, for the 64 m × 64 m section of the Alonsa, Manitoba, tornado shown in Fig. 10.

even with differing architectures, leading to the conclusion that the performance of the model is limited more by the training data than the size of model used. Overall, a Dice score of 95.1% indicates that the model can effectively extract almost all trees stems in the imagery. Figures 10–12 show empirical tests of automatically produced treefall segmentation masks for the Alonsa, Lac Gus, and Mary Lake tornado events.

Using both the tree semantic segmentation model and the instance segmentation algorithm, an automated tree tagging model is produced. Figure 13 shows an example result of the automated tree tagging model run on a section of the Alonsa tornado. The result performs effectively, fitting lines to almost all tree stems. However, several branches are picked up or lines not joined together due to the curvature of a tree. Additionally, a comparison of manual versus automatic tree tagging is performed on 100 m × 100 m sections of the Alonsa, Lac Gus, and Lac Des Iles tornadoes, an example of which is

shown in Fig. 14. The results, shown in Table 4, conclude that the automated model has a mean recall of 90.2%, precision of 72.5%, and Dice score of 80.3%. The recall is relatively high, meaning that the model can find almost all trees in each image, but the precision at 72.5% is less than ideal, showing that a significant portion of the model’s predictions are false positives on branches or other linear-shaped debris. Overall, the model achieves a Dice score of 80.3%, which is satisfactory for the end goal of automatically producing TrIDA maps.

The following equations show the formulas used to calculate precision, recall, and the Dice score for Table 4:

$$\text{precision} = \frac{T_P}{T_P + F_P} = \frac{T_P}{\beta}, \tag{1}$$

$$\text{recall} = \frac{T_P}{T_P + F_N} = \frac{T_P}{P}, \text{ and} \tag{2}$$

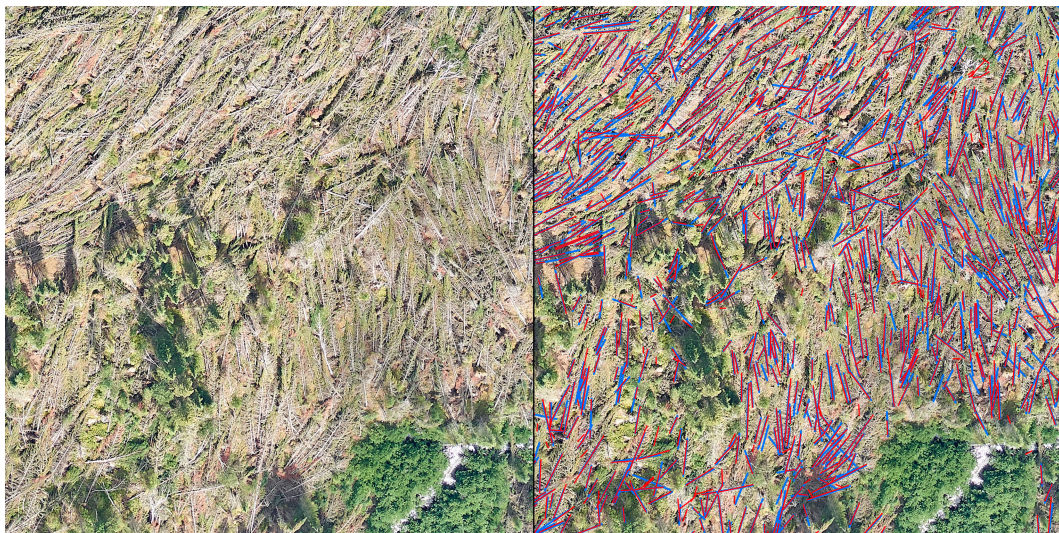


FIG. 14. (left) A 100 m × 100 m section of the Lac Gus, Quebec, tornado along with (right) manually (blue) and automatically (red) tagged tree lines.

TABLE 4. Results comparing manual to automatic tree instance segmentation for 100 m × 100 m sections. The “trees” column lists the number of real trees manually tagged. The “predictions” column lists the number of trees the algorithm predicted. The “found trees” column lists the number of trees correctly predicted. The bold font highlights the most important result.

Event	Trees	Predicted trees	Found trees	Precision	Recall	Dice (F1)
Alonsa	432	512	379	74.0%	87.8%	80.3%
Lac Gus	792	965	725	75.1%	91.5%	82.5%
Lac Des Iles	386	515	352	68.3%	91.2%	78.2%
Mean	$x$	$x$	$x$	72.5%	90.2%	<b>80.3%</b>

$$\text{Dice} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}, \quad (3)$$

where  $T_P$ ,  $F_P$ , and  $F_N$ , represent the true positives (model predicted a tree where there is a tree), false positives (model predicted a tree where there is not a tree or multiple trees where there is only one tree), and false negatives (model did not predict a tree where there is a tree), respectively. Moreover,  $\beta$  and  $P$  represent the total number of trees predicted by the automated tree tagging model, and total number of actual trees, respectively. Additionally, the mean precision and recall can also be used to produce an estimate for the ratio between the number of predicted trees  $\beta$  and the number of actual trees  $P$  in an image:

$$\beta \frac{\text{precision}}{\text{recall}} = P, \quad \text{and} \quad (4)$$

$$\frac{\text{precision}_{\text{mean}}}{\text{recall}_{\text{mean}}} \approx 0.80. \quad (5)$$

Table 5 shows the number of fallen trees predicted by the automated tree tagging model, followed by an estimate for the actual number of fallen trees for various events tested. The estimate for the actual number of fallen trees is calculated by multiplying the ratio in Eq. (5) by the number of predicted fallen trees tagged by the automated model. Only the trees in the tornadic damage path captured in the collected aerial imagery are considered, excluding any identified downburst damage.

Table 6 shows the results from training various tree direction models. The best model trained has a validation accuracy and Dice score of 86.6%, using a VGG-19 backbone followed by a 4096 node fully connected layer. Although not perfect, 86.6% is suitable for the end goal of producing a TrIDA map as fall directions will be considered over larger areas rather than each individual tree’s direction.

TABLE 5. Number of fallen trees predicted and estimates for the actual number of fallen trees from the tornadic damage of various events tested.

Tornado event	Number of fallen trees predicted	Estimated actual number of fallen trees (two significant figures)
Brooks Lake, Ontario	911 659	730 000
Alonsa, Manitoba	286 684	230 000
Lac Gus, Quebec	56 983	46 000

#### 4. Automated production of treefall vector maps

Following detection and direction predictions, the large-scale aerial images are split into a grid, with each tagged tree line’s unit vector sorted into a grid square based on the tagged line’s midpoint. From here, the unit vectors in each grid square are assessed using downsampling methods such as the mean, median, or clustering, to find the area-averaged direction for each grid square. In theory, any grid size could be chosen, but the larger the grid square, generally the more fallen trees will be present. Moreover, using a grid size which is too large will downsample too much data, leading to less useful results. As such, grid sizes are manually chosen relative to the size of the tornado damage path and desired resolution, usually between 25 and 250 m in size. Additionally, using the median is preferred over the mean due to it being more resistant to outliers.

However, there is a possibility of more than one distinct cluster of directions. This can be due to many factors, including multiple vortices in the tornado, a tornado changing course and hitting the same location multiple times, a combination of tornado and downburst damage, etc. Based on observations from manual analysis of treefall patterns, it is most likely to have one or two significant distinct clusters of treefall directions. To account for the possibility of two distinct clusters of directions in a grid square, a method of fitting two area-averaged directions per grid square is developed. First, the unit vectors in each grid square are assumed to be in two clusters, which are fit using the  $k$ -medoids clustering algorithm (Kaufman and Rousseeuw 1990) with  $k$  equal to 2. The medoid of a cluster is the data point that is the most similar to all other data points in the cluster, which, like the median, is more resistant to outliers. Given that the data points are unit vectors, the similarity between vectors is defined by comparing the polar angles of each vector. After clustering, the ratio of data points and the angle between the medoid vector of each cluster is considered. If the ratio is more balanced than a 60:40 split and the angle between the medoid vectors is greater than 45°, the two medoid vectors are utilized instead of the median vector, as shown in Fig. 15.

To assess the automated model’s performance, a comparison is made between the manually assessed treefall directions used in Stevenson et al. (2023) for the Alonsa tornado and the model’s automatically assessed directions. Both median and clustering methods with a grid size of 25 m are used with the automated model, and a relatively small example section using the clustering method can be seen in Fig. 16. The vectors are compared by calculating the difference between their angles and producing histograms with bin sizes of 5°, as shown

TABLE 6. Training results for various tree direction convolutional neural networks. The bold font indicates the best model.

Convolutional backbone architecture	Fully connected layer size	Learning rate	Batch size	Validation accuracy	Validation dice (F1)
ResNet-50	2048	0.000 05	32	83.3%	83.2%
	4096	0.000 05	16	82.9%	82.9%
VGG-16	2048	0.000 05	32	85.8%	85.8%
	4096	0.000 03	16	84.8%	84.5%
VGG-19	2048	0.000 05	16	86.1%	86.1%
	<b>4096</b>	<b>0.000 01</b>	<b>16</b>	<b>86.6%</b>	<b>86.6%</b>

in Fig. 17. To compare the clustering method, if two vectors are produced for a given grid square, the vector whose angle is closer to the manual vector is considered. This is done to account for the fact that even in a 25-m grid square, there can be two sets of distinct directions, only one of which is chosen during manual assessment.

The treefall directions used in Stevenson et al. (2023) are the result of the semiautomated treefall extraction method used in Rhee et al. (2021), followed by significant manual adjustment. For our comparison, some vectors between the manual and automated model do not align perfectly. To account for this, the automatically generated vectors are compared to the nearest neighboring manual vector. A grid size of 25 m is used, resulting in a worst-case distance between manual and automatic vectors of  $\sqrt{2} \times 12.5 \text{ m} \approx 17.68 \text{ m}$ , though generally less, as can be seen in Fig. 16.

As shown in Fig. 17, the distributions of angle differences are exponentially decreasing, with a median value of approximately 13.3° for both the median and clustering methods. Moreover, the clustering method performs better than the median method in terms of both mean and 80% values demonstrating its viability. Angle differences between 130° and 180° tend to occur as error on the part of model but was reduced significantly with the clustering method as shown in Fig. 17. Angle differences of 30°–80° tend to occur closer to where the trees converge toward the center of the tornado’s path as there can be rapid changes in direction leading to

inconsistency between manual and automated assessment. Overall, when considering the inconsistent alignment between manual and automatic vectors and factoring in human error when performing a manual assessment, a median of 13.3° and 80% threshold of 25.3° demonstrates effective results.

The appendix provides a step-by-step process to how the automated treefall model is implemented for generating TriDA maps, along with a visual representation of the similarity between manual and automated analyses.

### 5. Conclusions

Around the world, many tornadoes occur in unpopulated forested regions. To more accurately analyze the aerial imagery obtained from these events, Treefall Identification and Direction Analysis (TriDA) maps are developed to analyze treefall damage. An automated machine learning method for the detection and recording of fallen trees and their fall directions from large-scale aerial imagery is implemented in order to reduce the time and labor of producing TriDA maps. The automated method first uses a U-Net-based image segmentation model to extract fallen tree-stem pixels from the aerial imagery. Next, a constructed algorithm is used to perform instance segmentation, identifying and separating the individual fallen trees. Then, a convolutional neural network is used to assess the directions from images of each individual tree. Last, the

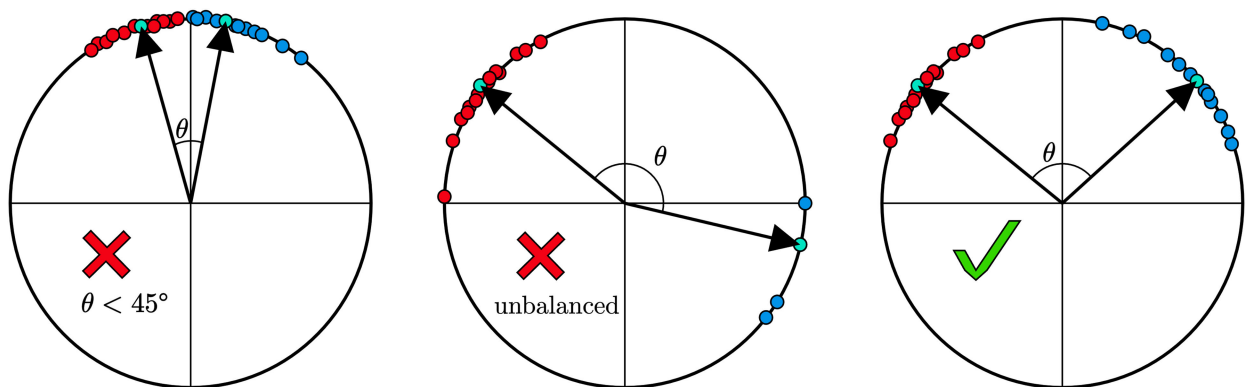


FIG. 15. Depiction of the clustering algorithm utilized to find two distinct clusters of directions in a grid square. The dots represent the endpoints of unit vectors placed on the unit circle. The red and blue dots show the corresponding cluster, and the teal dots represent the medoid of each cluster.

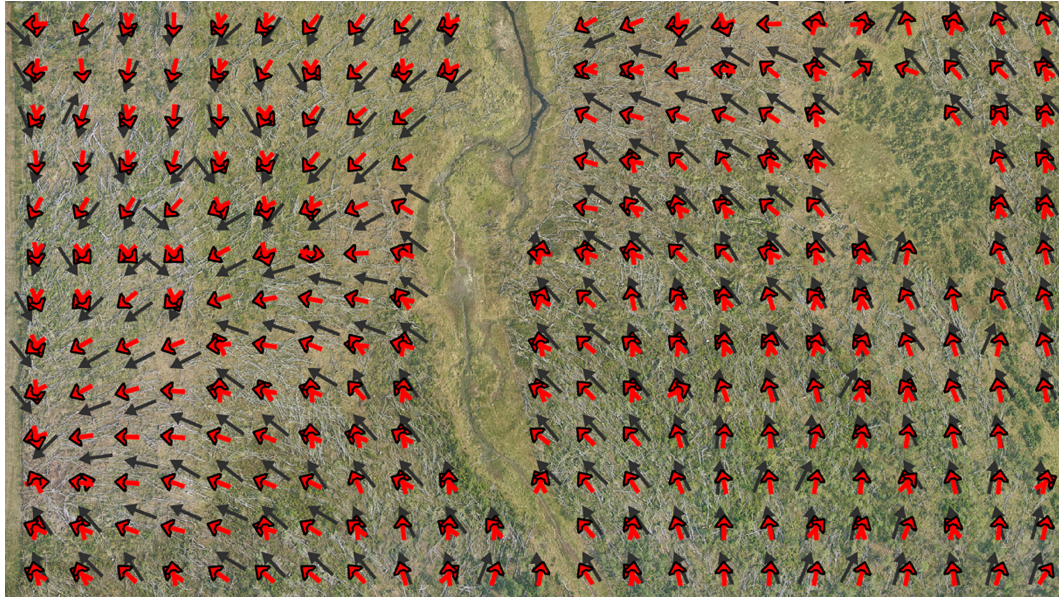


FIG. 16. Manual (black) vs automated (red) treefall direction field with a 25-m grid size for a 575 m  $\times$  325 m section of the Alonsa, Manitoba, tornado aerial imagery.

imagery is split into a grid, and the area-averaged direction of all the trees in each grid square is assessed using the median or  $k$ -medoids clustering algorithm to produce a treefall vector map.

The U-Net-based treefall semantic segmentation model achieved a validation Dice score of 95.1%, indicating that the model can effectively extract almost all tree stems from the aerial imagery. When compared to manually detected trees, the automated model has a mean tree detection Dice score of 80.3%, which is satisfactory for the end goal of automatically producing treefall vector maps. The automated model achieves a median tree direction difference of 13.3° when compared to

the manual tree directions from the Alonsa, Manitoba, tornado, demonstrating the viability of the automated model compared to manual assessment. Overall, the automated production of treefall vector maps from large-scale aerial imagery significantly speeds up and reduces the labor required to create a Treefall Identification and Direction Analysis map going from a matter of days or weeks to a matter of hours.

All model training and testing was performed on a workstation with an Intel i9-12900K CPU, Nvidia RTX 3080 10GB GPU, 64 GB of DDR5 RAM, and a Samsung 980 pro NVME SSD. The time to perform semantic and then instance

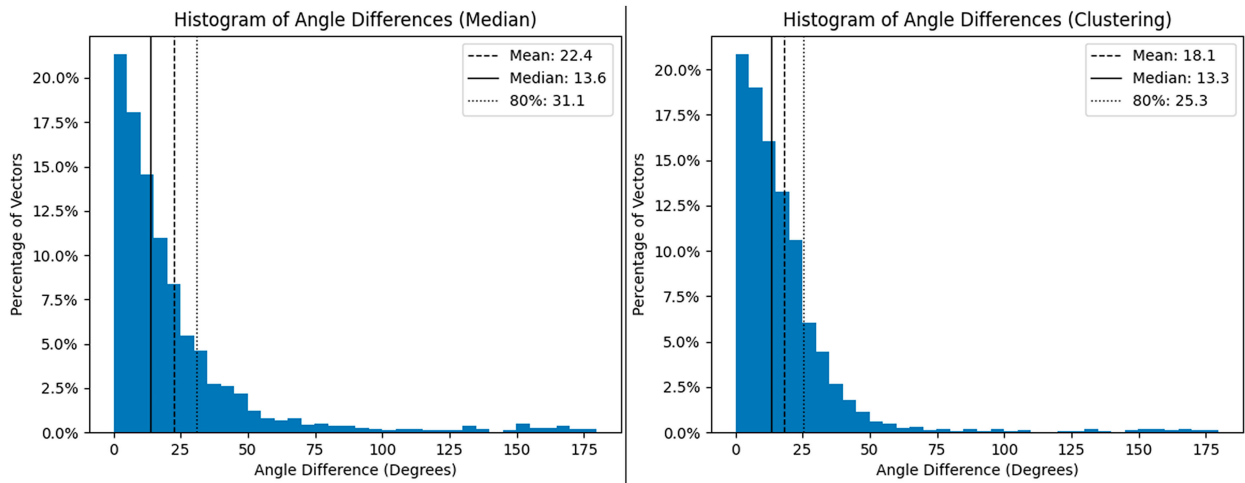


FIG. 17. Histograms comparing the angle difference between manual and automated treefall direction vectors for the Alonsa, Manitoba, tornado. Bins of 5° are used comparing both median and clustering methods. The 80% mark indicates the value at which 80% of the vectors have been counted.

segmentation takes approximately 1–3 min per  $20\,000 \times 20\,000$  pixel  $1\text{-km}^2$  image depending on the number of trees in each image. The time to determine the treefall directions and produce an area-averaged treefall vector map is an additional 1–2 min per image depending on the number of trees in each image and the averaging method selected. As an example, the imagery for the Alonsa, Manitoba, tornado contained thirty-seven  $20\,000 \times 20\,000$  pixel  $1\text{-km}^2$  images and took just under 2 h to complete. For comparison, the same process of identifying all treefall directions manually would typically take weeks.

The model requires imagery that is of a high enough resolution and quality to clearly distinguish between trees, generally  $5\text{ cm} \times 5\text{ cm}$  per pixel or better, which currently limits the model to using imagery collected from aircraft or UAVs. As the resolution of satellite imagery improves in the future to the point where individual trees and their treefall direction can be resolved (reducing the reliance on expensive aircraft or UAVs), this method will become a relatively inexpensive, globally available way to analyze treefall damage from tornadoes. Currently, no commercially available satellite imagery is capable of this high resolution. Additionally, if aircraft or UAV imagery improves in the future, this method could potentially be applied to analyzing crop damage, in the manner demonstrated in Baker et al. (2020). The higher resolution is required to detect the thin stems of the crops. An additional limitation is the use of the model in forest areas surrounding or incorporated into urban areas. Often the model incorrectly identifies many brightly colored linearly shaped objects including road lines, roof edges, and wooden fences, as trees.

Future improvements will be made to both the treefall semantic segmentation and direction models by adding additional imagery from other tornadic events in order to increase the size and diversity of the training datasets. Moreover, improvements are planned to design a more robust tree instance segmentation algorithm with a higher precision, and Dice score. Additionally, automated detection and segmentation of damaged areas, along with the automated fitting of tornado convergence lines will also be examined in the future.

*Acknowledgments.* This research was funded by ImpactWX and Western University. The authors would like to acknowledge the contributions of Dr. Daniel Rhee, Dr. Frank Lombardo, and Mr. Emilio Hong for their assistance with comparing the automated model to the manually tagged Alonsa, Manitoba, tornado.

*Data availability statement.* All the open access data used in this study can be downloaded from the Northern Tornadoes Project website ([www.uwo.ca/ntp](http://www.uwo.ca/ntp)). An ArcGIS Pro add-on along with the code used in the study can be found in our GitHub repository ([https://github.com/Daniel-Butt/Tree\\_Tagger\\_ArcGIS\\_Add\\_On](https://github.com/Daniel-Butt/Tree_Tagger_ArcGIS_Add_On)).

## APPENDIX

### Implementing the Automated Treefall Model for TrIDA Maps

The following step-by-step process describes how to generate a TrIDA map with the assistance of the automated model, starting after the acquisition and processing of the aerial orthomosaic imagery.

- 1) Identify all the treefall damage in the imagery by drawing contours around the areas with fallen trees that appear to be from the severe wind event being analyzed. Note that it is possible to perform this step using the automated model, but performing the task manually results in more precise contours and fewer false positives such as fallen trees unrelated to the severe wind event being analyzed.
- 2) Based on the damage contours, identify the damage centerline, maximum width, and worst box of damage along the damage path.
- 3) Use the automated model to identify all fallen trees in the imagery, as well as their direction.
- 4) Make any manual edits necessary to the automatically identified trees. This involves deleting false positives, which primarily consist of mistakenly identified trees from features such as buildings and roads.
- 5) Use the automated model to produce area-averaged treefall directions using one or several specified area-averaging methods and grid sizes. When multiple distinct clusters of treefall directions are observed in a grid space, this will be represented as two overlapping area-averaged treefall vectors.
- 6) If necessary, manually adjust any inaccurate area-averaged treefall directions.
- 7) Using the automated area-averaged treefall directions as a guide, identify areas of tornadic and straight-line wind damage.
- 8) Using the automated area-averaged tree directions as a guide, manually draw the tornado convergence line, if applicable.

A visual comparison between TrIDA maps produced manually and with the automated treefall vector field can be seen in Figs. A1–A3. The important takeaway from these figures is that the TrIDA maps generated with the help of the automated model produce equally accurate and more organized treefall directions, while significantly reducing the manual time and labor otherwise required to generate these maps without the help of the automated model. Including any pre-processing of imagery and manual adjustments needed to generate a final TrIDA map, this whole process typically only takes 1–2 days, whereas generating a complete TrIDA map manually often takes a week or more.





FIG. A1. (left) Manual and (right) automated TrIDA maps for a  $4.25 \text{ km} \times 4.25 \text{ km}$  section of the Brooks Lake, Ontario, tornado with a 250-m grid size. This section of the Brooks Lake tornado does not overlap with any images used for training or validation of the automated model.



FIG. A2. (left) Manual and (right) automated TrIDA maps for a  $2.6 \text{ km} \times 2.6 \text{ km}$  section of the Alonsa, Manitoba, tornado with a 120-m grid size.

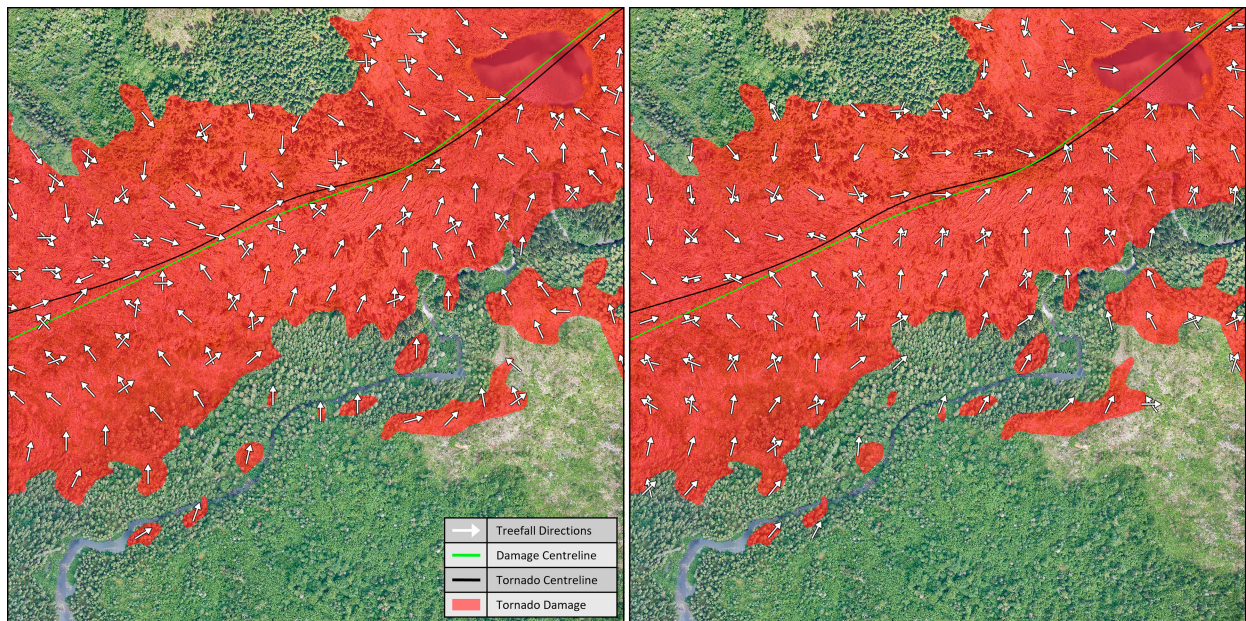


FIG. A3. (left) Manual and (right) automated TrIDA maps for a  $600\text{ m} \times 600\text{ m}$  section of the Lac Gus, Quebec, tornado with a 40-m grid size.

#### REFERENCES

- Abadi, M., and Coauthors, 2016: TensorFlow: A system for large-scale machine learning. *Proc. 12th USENIX Symp. on Operating Systems Design and Implementation (OSDI'16)*, Savannah, GA, USENIX Association, 265–283, <https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf>.
- Baker, C., M. Sterling, and M. Jesson, 2020: The lodging of crops by tornadoes. *J. Theor. Biol.*, **500**, 110309, <https://doi.org/10.1016/j.jtbi.2020.110309>.
- Breiman, L., 2001: Random forests. *Mach. Learn.*, **45**, 5–32, <https://doi.org/10.1023/A:1010933404324>.
- Chollet, F., and Coauthors, 2015: Keras. <https://keras.io>.
- Dice, L. R., 1945: Measures of the amount of ecologic association between species. *Ecology*, **26**, 297–302, <https://doi.org/10.2307/1932409>.
- Duan, F., Y. Wan, and L. Deng, 2017: A novel approach for coarse-to-fine windthrown tree extraction based on unmanned aerial vehicle images. *Remote Sens.*, **9**, 306, <https://doi.org/10.3390/rs9040306>.
- Fujita, T. T., 1989: The Teton-Yellowstone tornado of 21 July 1987. *Mon. Wea. Rev.*, **117**, 1913–1940, [https://doi.org/10.1175/1520-0493\(1989\)117<1913:TTYTOJ>2.0.CO;2](https://doi.org/10.1175/1520-0493(1989)117<1913:TTYTOJ>2.0.CO;2).
- , and R. M. Wakimoto, 1981: Five scales of airflow associated with a series of downbursts on 16 July 1980. *Mon. Wea. Rev.*, **109**, 1438–1456, [https://doi.org/10.1175/1520-0493\(1981\)109<1438:FSOAAW>2.0.CO;2](https://doi.org/10.1175/1520-0493(1981)109<1438:FSOAAW>2.0.CO;2).
- Godfrey, C. M., and C. J. Peterson, 2017: Estimating enhanced Fujita scale levels based on forest damage severity. *Wea. Forecasting*, **32**, 243–252, <https://doi.org/10.1175/WAF-D-16-0104.1>.
- Government of Canada, 2020: ISO 19131 population of Canada, 10 km gridded. Accessed 30 March 2023, <https://open.canada.ca/data/en/dataset/c6c48391-fd2f-4d8a-93c8-eb74f58a859b>.
- Guo, Z., and R. W. Hall, 1992: Fast fully parallel thinning algorithms. *CVGIP Image Understanding*, **55**, 317–328, [https://doi.org/10.1016/1049-9660\(92\)90029-3](https://doi.org/10.1016/1049-9660(92)90029-3).
- He, K., X. Zhang, S. Ren, and J. Sun, 2016: Deep residual learning for image recognition. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Las Vegas, NV, Institute of Electrical and Electronics Engineers, 770–778, <https://doi.org/10.1109/CVPR.2016.90>.
- Hough, P. V. C., 1962: Method and means for recognizing complex patterns. Office of Scientific and Technical Information, U.S. Department of Energy, U.S. Patent 3069654, <https://www.osti.gov/doi/patents/biblio/4746348>.
- Karra, K., C. Kontgis, Z. Statman-Weil, J. C. Mazzariello, M. Mathis, and S. P. Brumby, 2021: Global land use/land cover with Sentinel 2 and deep learning. *2021 IEEE Int. Geoscience and Remote Sensing Symp. IGARSS*, Brussels, Belgium, Institute of Electrical and Electronics Engineers, 4704–4707, <https://doi.org/10.1109/IGARSS47720.2021.9553499>.
- Karstens, C. D., W. A. Gallus Jr., B. D. Lee, and C. A. Finley, 2013: Analysis of tornado-induced tree fall using aerial photography from the Joplin, Missouri, and Tuscaloosa–Birmingham, Alabama, tornadoes of 2011. *J. Appl. Meteor. Climatol.*, **52**, 1049–1068, <https://doi.org/10.1175/JAMC-D-12-0206.1>.
- Kaufman, L., and P. J. Rousseeuw, 1990: Partitioning around medoids (program PAM). *Finding Groups in Data: An Introduction to Cluster Analysis*, Wiley Series in Probability and Statistics, Vol. 344, John Wiley and Sons, 68–125, <https://doi.org/10.1002/9780470316801.ch2>.
- Kingma, D. P., and J. Ba, 2014: Adam: A method for stochastic optimization. arXiv, 1412.6980v9, <https://doi.org/10.48550/arXiv.1412.6980>.
- McDonald, J., and K. C. Mehta, 2006: A recommendation for an enhanced Fujita scale (EF-scale). Wind Science and Engineering Center, Texas Tech University, 95 pp., <https://www.spc.noaa.gov/faq/tornado/ef-ttu.pdf>.
- Mehta, K., 2013: Development of the EF-scale for tornado intensity. *J. Disaster Res.*, **8**, 1034–1041, <https://doi.org/10.20965/jdr.2013.p1034>.

- Reder, S., J.-P. Mund, N. Albert, L. Waßermann, and L. Miranda, 2022: Detection of windthrown tree stems on UAV-orthomosaics using U-Net convolutional networks. *Remote Sens.*, **14**, 75, <https://doi.org/10.3390/rs14010075>.
- Rhee, D. M., and F. T. Lombardo, 2018: Improved near-surface wind speed characterization using damage patterns. *J. Wind Eng. Ind. Aerodyn.*, **180**, 288–297, <https://doi.org/10.1016/j.jweia.2018.07.017>.
- , —, and J. Kadowaki, 2021: Semi-automated tree-fall pattern identification using image processing technique: Application to Alonsa, MB tornado. *J. Wind Eng. Ind. Aerodyn.*, **208**, 104399, <https://doi.org/10.1016/j.jweia.2020.104399>.
- Ronneberger, O., P. Fischer, and T. Brox, 2015: U-Net: Convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015*, N. Navab et al., Eds., Lecture Notes in Computer Science, Vol. 9351, Springer, 234–241, [https://doi.org/10.1007/978-3-319-24574-4\\_28](https://doi.org/10.1007/978-3-319-24574-4_28).
- Sills, D. M. L., P. J. McCarthy, and G. A. Kopp, 2014: Implementation and application of the EF-scale in Canada. *27th Conf. on Severe Local Storms*, Madison, WI, Amer. Meteor. Soc., 16B.6, <https://ams.confex.com/ams/27SLS/webprogram/Paper254999.html>.
- , and Coauthors, 2020: The Northern Tornadoes Project: Uncovering Canada’s true tornado climatology. *Bull. Amer. Meteor. Soc.*, **101**, E2113–E2132, <https://doi.org/10.1175/BAMS-D-20-0012.1>.
- Simonyan, K., and A. Zisserman, 2014: Very deep convolutional networks for large-scale image recognition. arXiv, 1409.1556v6, <https://doi.org/10.48550/arXiv.1409.1556>.
- Snoek, J., H. Larochelle, and R. P. Adams, 2012: Practical Bayesian optimization of machine learning algorithms. *Advances in Neural Information Processing Systems 25 (NIPS 2012)*, F. Pereira et al., Eds., Curran Associates, 2951–2959.
- Stevenson, S. A., C. S. Miller, D. M. L. Sills, G. A. Kopp, D. M. Rhee, and F. T. Lombardo, 2023: Assessment of wind speeds along the damage path of the Alonsa, Manitoba EF4 tornado on 3 August 2018. *J. Wind Eng. Ind. Aerodyn.*, **238**, 105422, <https://doi.org/10.1016/j.jweia.2023.105422>.
- von Gioi, R. G., J. Jakubowicz, J.-M. Morel, and G. Randall, 2010: LSD: A fast line segment detector with a false detection control. *IEEE Trans. Pattern Anal. Mach. Intell.*, **32**, 722–732, <https://doi.org/10.1109/TPAMI.2008.300>.