remote sensing

MDPI

Article

Using Deep Learning and Advanced Image Processing for the Automated Estimation of Tornado-Induced Treefall

Mitra Nasimi and Richard L. Wood *

Department Civil and Environmental Engineering, University of Nebraska-Lincoln, Lincoln, NE 68588, USA; mnasimi2@huskers.unl.edu
* Correspondence: rwood@unl.edu

Abstract: Each year, numerous tornadoes occur in forested regions of the United States. Due to the substantial number of fallen trees and accessibility issues, many of these tornadoes remain poorly documented and evaluated. The process of documenting tree damage to assess tornado intensity is known as the treefall method, an established and reliable technique for estimating near-surface wind speed. Consequently, the demand for documenting fallen trees has increased in recent years. However, the treefall method proves to be extremely expensive and time-consuming, requiring a laborious assessment of each treefall instance. This research proposes a novel approach to evaluating treefall in large, forested regions using deep learning-based automated detection and advanced image processing techniques. The developed treefall method relies on high-resolution aerial imagery from a damaged forest and involves three main steps: (1) instance segmentation detection, (2) estimating tree taper and predicting fallen tree directions, and (3) obtaining subsampled treefall vector results indicating the predominant flow direction in geospatial coordinates. To demonstrate the method's effectiveness, the algorithm was applied to a tornado track rated EF-4, which occurred on 10 December 2021, cutting through the Land Between the Lakes National Recreation Area in Kentucky. Upon observation of the predicted results, the model is demonstrated to accurately predict the predominant treefall angles. This deep-learning-based treefall algorithm has the potential to speed up data processing and facilitate the application of treefall methods in tornado evaluation.

Keywords: tornado damage; treefall; damage assessment; deep learning; aerial imagery

check for updates

Citation: Nasimi, M.; Wood, R.L. Using Deep Learning and Advanced Image Processing for the Automated Estimation of Tornado-Induced Treefall. *Remote Sens.* **2024**, *16*, 1130. https://doi.org/10.3390/rs16071130

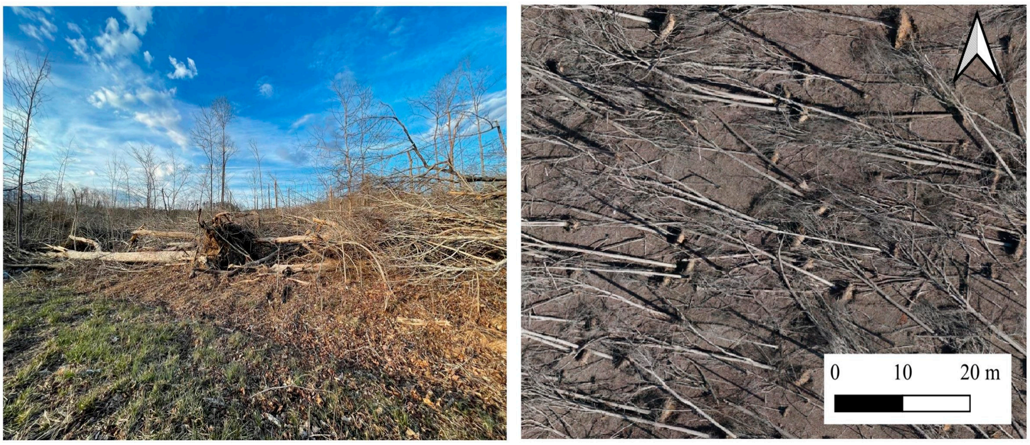
Academic Editors: Weifeng Liu, Igor García Olaizola and Bingfeng Zhang

Received: 29 January 2024
Revised: 8 March 2024
Accepted: 21 March 2024
Published: 23 March 2024

1. Introduction

Tornados are among the most destructive weather phenomena throughout the world. In 2022, more than one thousand tornadoes occurred only across the United States [1], where the damage caused by tornadoes reached more than $708 million [2]. Performance-based engineering philosophies have been recently developed to reduce these losses associated with the low probability of occurrence but with high consequence tornadic events (e.g., [3,4]). However, the quantification of tornado hazard is still ongoing and requires an estimation of the near-surface wind speeds. Unfortunately, there are no direct methods to measure the majority of near-surface winds, where the strongest winds occur near the ground level [5] and where the people and anthropogenic and natural objects reside (i.e., buildings and trees). Consequently, post-storm evaluation is the most common method in evaluating tornado intensity. Figure 1 shows two different types of data used for the post-storm evaluations.

Post-tornado storm evaluations can become more challenging, as a large number of tornadoes occur in forested rural areas with low density of anthropogenic structures. The Enhanced Fujita scale is a method developed to estimate the wind speeds of tornadoes. This method is utilized according to anthropogenic damage indicators (Dis). It is critical to note that the current Enhanced Fujita (EF) method is unreliable in estimating wind speeds in regions lacking human-made features and buildings. To evaluate forest damage due to windstorms, previous research has focused on understanding the damage mechanism of the trees to determine the critical wind effect on the trees (e.g., [6]). This research identified individual tree responses and associated damage following extreme winds. However, in reality, the resistance of an individual tree is influenced by the interaction of the forested trees with each other. This research identified individual tree responses and associated damage caused by extreme winds. However, it is important to consider that in a forest, the interactions among trees significantly influence the resistance of each tree. Consequently, assessing an individual tree's resistance alone does not provide a robust method for estimating the peak wind speeds in strong tornadoes. A computer model of the forest stand has been developed to address this. This model simulates the interactions between trees and wind, offering a deeper understanding of how tree density affects the overall resistance of the stand [7].



(**a**) (**b**)

Figure 1. Example of tornado-induced treefall: (**a**) ground-based image and (**b**) aerial image.

Tree damage is not solely a function of tree density, species, and wind speed. In addition, tree damage, along with tornado intensity, is related to terrain and location. This includes terrain effects such as surface roughness, elevation, and topography variation. In this regard, some computer simulations have considered the effect of a single parameter, such as surface roughness, as dominant in their modeling and evaluated variation of the roughness in the tornado intensification [8]. On the other hand, other computational modeling work includes topography and roughness [9,10]. While previous research studies provide valuable information about the tornado structure and their corresponding damage due to terrain, assumptions in generalized terrain effects due to computational demands and lack of data availability do not reflect the real-world interactions, limiting the scope of fully understanding tornado hazards.

Additionally, information on tornadoes in rural forested areas is limited due to a lack of visual confirmation and sparse reporting/documentation in these regions. However, unlike other natural hazards and in built-up anthropogenic regions, evidence of tornado damage is persistent in forested regions. Documenting tree damage to evaluate tornado intensity is known as the treefall method [11,12]. This is one of the most popular methods of tornado assessment in rural areas. Treefall data from nine forested regions were analyzed to explore the potential relationship between the angle of fallen trees and their diameter. The analysis revealed that the tree's diameter and its wind resistance can influence the angle of the fallen trees [13]. These research findings underscore the significance of on-site damage documentation following tornado events. However, this method, reliant on in-situ access to the damaged region, underscores the necessity for remote data collection to enhance efficiency and scalability.

Aerial imagery offers an efficient method for identifying and documenting tornado damage, concurrently addressing accessibility challenges in forested regions. This holds

Remote Sens. **2024**, *16*, 1130 2 of 24

Post-tornado storm evaluation is [illegible], and a large number of tornadoes occur in forested or rural areas with low density of anthropogenic structures. The Enhanced Fujita scale is a method developed to estimate the wind speeds of tornadoes. This method is utilized according to anthropogenic damage indicators (DIs). It is critical to note that the current Enhanced Fujita (EF) method is unreliable in estimating wind speeds in regions lacking human-made features and buildings. To evaluate forest damage due to windstorms, previous research has focused on understanding the damage mechanism of the trees to determine the critical wind effect on the trees (e.g., [6]). This research identified individual tree responses and associated damage following extreme winds. However, in reality, the resistance of an individual tree is influenced by the interaction of the forested trees with each other. This research identified individual tree responses and associated damage caused by extreme winds. However, it is important to consider that in a forest, the interactions among trees significantly influence the resistance of each tree. Consequently, assessing an individual tree's resistance alone does not provide a robust method for estimating the peak wind speeds in strong tornadoes. A computer model of the forest stand has been developed to address this. This model simulates the interactions between trees and wind, offering a deeper understanding of how tree density affects the overall resistance of the stand [7].

Tree damage is not solely a function of tree density, species, and wind speed. In addition, tree damage, along with tornado intensity, is related to terrain and location. This includes terrain effects such as surface roughness, elevation, and topography variation. In this regard, some computer simulations have considered the effect of a single parameter, such as surface roughness, as dominant in their modeling and evaluated variation of the roughness in the tornado intensification [8]. On the other hand, other computational modeling work includes topography and roughness [9,10]. While previous research studies provide valuable information about the tornado structure and their corresponding damage due to terrain, assumptions in generalized terrain effects due to computational demands and lack of data availability do not reflect the real-world interactions, limiting the scope of fully understanding tornado hazards.

Additionally, information on tornadoes in rural forested areas is limited due to a lack

particular significance given that not all tornadoes are documented due to underreporting [14]. One of the principal challenges lies in pinpointing tornado locations in sparsely populated areas. In addressing this challenge, researchers conducted a comprehensive analysis of satellite imagery data, utilizing various techniques to accurately detect the location and path information of tornado damage [15].

Satellite imagery data, with a resolution of 10 m, along with national land cover data at a resolution of 30 m, provides an opportunity to assess elevation and tornado intensity across the entire tornado by analyzing pixel intensity in various spectral bands, such as the Normalized Difference Vegetation Index (NDVI) [16]. However, even with commercial satellite imagery data, the current highest resolution is around 3 m, which proves generally insufficient to discern fallen trees. This limitation underscores the growing preference for aerial imagery, providing a more detailed view of the scene. For example, early studies utilized aerial imagery to identify distinctive damage patterns [17], document tornado marks [18], and integrate this information with topographic maps to establish a correlation between elevation and tornado damage in forests [19,20]. Additionally, when the resolution of the imagery is sufficient (25 cm), a tree fall pattern map can be constructed [21,22]. Ultimately, these maps can be used to develop near-surface wind speed through generated wind direction maps [12].

Additionally, high-resolution aerial imagery data with a resolution range of approximately 2–5 cm provides the opportunity to assess tornadoes with improved efficiency and scalability. For instance, pixel assessment techniques using the Visible Difference Vegetation Index (VDVI) can be employed to create an efficient colorized damage map [23]. Additionally, this finer resolution range provides the opportunity to estimate the EF scale from the percentage of fallen trees [11] and to compare the Enhanced Fujita scale (EF) damage indicator and the degree of damage with tree damage [24]. However, automated and semi-automated methods are sought to increase the efficiency of the assessment as well as to reduce human subjectivity.

In recent years and with advancements in computing, artificial intelligence (AI) encompassing both machine learning and deep learning techniques are increasingly used to analyze images. This directly aids in the creation of automated and semi-automated data analysis techniques. One of the earlier works that used deep learning to classify damaged regions utilized the 3D point cloud of windstorm-damaged areas [25]. However, this study demonstrated the computationally extensive and expensive nature of analyzing 3D data, with only a modest gain in the use of 3D versus 2D data [26]. Consequently, 2D imagery datasets are the most common datasets used for treefall analysis, along with other remote sensing analyses of tornado damage.

Prior research has shown that image processing techniques using aerial imagery can identify treefall patterns [27], with validation confirmed on three other popular techniques [28]. However, it was noted that the data were cumbersome and that automated processes could significantly enhance treefall methods. Consequently, deep learning and machine learning approaches have been utilized to address this challenge. For example, AI categorized images based on tornado damage and generated a heat damage map [29]. Various AI techniques demonstrated high proficiency in analyzing 2D images for tornado damage detection. In general, the findings indicated superior performance of deep learning methods over machine learning [30], potentially due to the complexity of the data. Furthermore, deep learning methods utilizing a pre-trained generic dataset demonstrated improved accuracy in identifying damaged regions and locating fallen trees [31]. However, a primary limitation of these methods lies in the low amount of training data, which results in limited model generalization.

Aiming to increase the accuracy of the estimated near surface-wind speed of the tornado, previous studies have developed methods and algorithms to evaluate damage along the tornado path. However, many studies have not taken into consideration some limitations related to human subjectivity and bias, assessing the whole tornado path due to extensive labor work and processing time. Another limitation of those methods is the accessibility to the high-quality data set. This research aims to provide a time-efficient automated method empowered by deep learning and image processing techniques to identify the direction of the fallen trees using high-resolution aerial imagery data to generate the wind direction map in the impacted area. One of the main advantages of the proposed method is the high processing speed and generalizability of the trained model, in addition to improved objectivity.

To this end, this paper aims to overcome the limitations identified in previous research regarding the estimation of treefall angles and the generation of wind direction maps. It introduces significant contributions to the field, with key factors including the application of an efficient deep-learning method for improved scalability. This enables the proposed method to detect a higher number of fallen trees effectively. Moreover, the algorithm integrates tree taper to achieve accurate estimations of tornado-induced fallen tree directions, enhancing the interpretability of the deep learning model. Additionally, a parametric analysis is conducted to eliminate inaccurate predictions by considering the physical properties of the tree annotations and their deviations. Ultimately, the methodology goes beyond conventional approaches by generating a wind direction map based on the averaged treefall damage. These contributions collectively strengthen the robustness and precision of the proposed methodology.

2. Dataset and Data Preparation

The data used in this project and the development of the algorithm consist of post-tornado aerial imagery obtained from an uncrewed aerial system (UAS) in a forested region of Kentucky and Tennessee. This forest region was devoid of many anthropogenic structures, except for a few recreational sites and access roads. The forest inventory primarily consisted of deciduous trees, but conifer trees were also present. The aerial imagery used in the model development comprises high-resolution orthomosaic images in the visible spectrum of approximately 2.0 cm, tiled at 1024 square pixels. The format of the tiled images is GeoTIFF, which includes geospatial information in terms of geographic coordinates and projection details.

These images were generated using a Structure-from-Motion (SfM, version 2.0.2) software platform [32]. Here, orthomosaic images are desirable due to their nadir view, corrected for perspective, camera angle, lens distortion, and topographic relief. They were collected using a fixed-wing UAS platform in a single-grid flight pattern, including post-processing kinematic tagging for centimeter accuracy precision [33]. Moreover, the images were collected during the leaf-off season to allow the camera view to penetrate the tree canopy. The data is further described in Section 4, where the model is applied to determine the near-surface wind direction map for a large region of a tornado track through a forest.

3. Methodology

This research focuses on analyzing and vectorizing treefall patterns through an aerial imagery dataset in the visible spectrum. The algorithm crafted for this purpose consists of three primary phases to identify treefall patterns. Initially, the dataset undergoes pre-processing, training a deep learning model to spot fallen tree trunks. Following this, the imagery dataset is processed to identify fallen trees in each image. The third phase involves image processing to vectorize the direction of the fallen tree and geolocate treefall angles.

Figure 2 illustrates a summary flowchart of the algorithm.

In the developed deep learning model, the output is the instance segmentation of each treefall, providing the boundaries of the tree trunk in pixel coordinates. Subsequently, the polygon coordinates are utilized to quantify the tree taper rate for each tree trunk. Tree taper, a critical factor in determining the direction of a fallen tree, is defined as the reduction in tree diameter with the increase in the tree's height [34]. This step is critical due to the rectangular nature of the detected tree trunks, ensuring the correct orientation from the root to the top of the tree. Finally, the treefall directions are converted to their geolocation and, if needed, subsampled to generate a representative treefall pattern.

quently, the polygon coordinates are utilized to quantify the tree taper rate for each tree trunk. Tree taper, a critical factor in determining the direction of a fallen tree, is defined as the reduction in tree diameter with the increase in the tree's height [34]. This step is critical due to the rectangular nature of the detected tree trunks, ensuring the correct orientation from the root to the top of the tree. Finally, the treefall directions are converted to their geolocation and, if needed, subsampled to generate a representative treefall pattern.

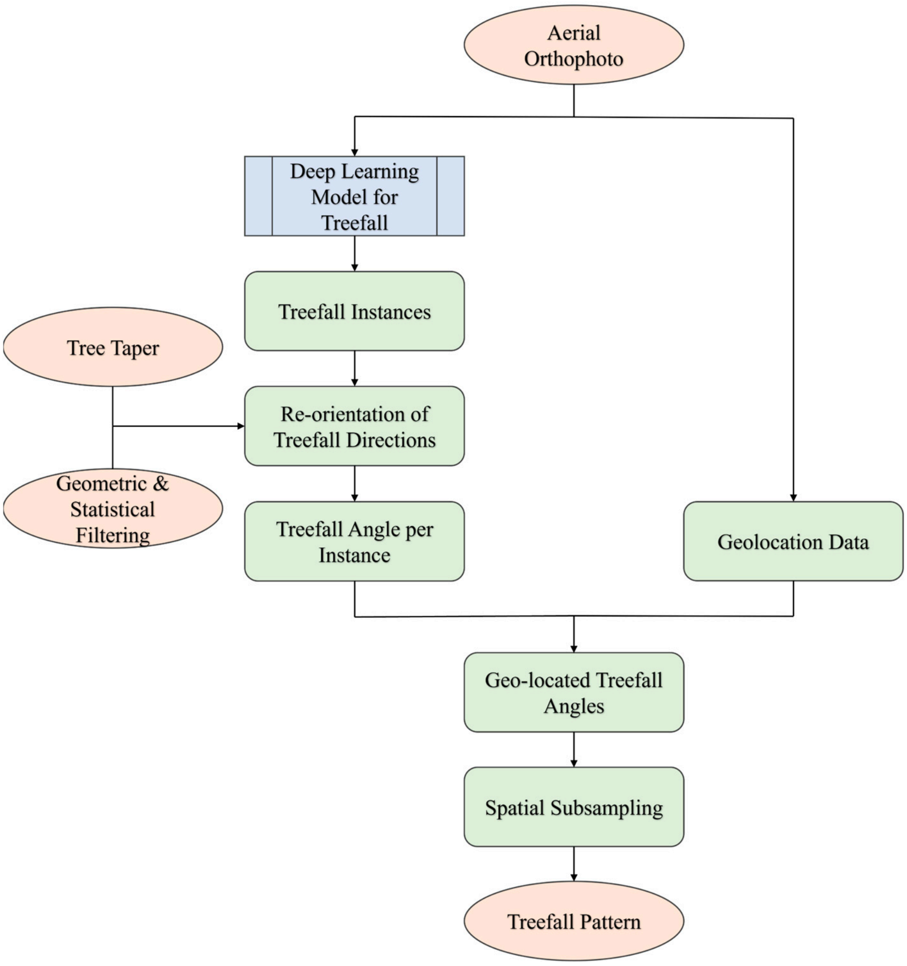

Aerial Orthophoto
Deep Learning Model for Treefall
Treefall Instances
Tree Taper
Re-orientation of Treefall Directions
Geometric & Statistical Filtering
Treefall Angle per Instance
Geolocation Data
Geo-located Treefall Angles
Spatial Subsampling
Treefall Pattern

Figure 2. Synopsis flowchart of the developed treefall algorithm.

3.1.2. DL Training Model

In the developed method, the workflow initiates with tiled orthomosaic images in the visible spectrum. The images in this study were annotated for two categories: treefall and root ball instances. In the segmentation of treefall instances, focus is placed exclusively on the visible portion of the tree trunk. This includes the area from the base or the lowest point of the tree, extending to where a deciduous tree branches out or encompassing the predominant length of conifer trees. It is important to note that while trees comprise branches and roots, these components are not always visible or discernible in the dataset. This lack of visibility is particularly common in densely forested areas, where treetops often overlap, making them unclear. The annotations were performed by one person, and a second person always performed quality control validation [35]. The annotated images train the selected YOLOv8x-seg model [36]. This is the large instance segmentation-specific version of YOLOv8x-seg developed by Ultralytics, which stands for "You Only Look Once-level version 8". The series of YOLOv8 models and their predecessors are known for their real-time object detection capabilities that utilize single-shot detection [37]. This is a critical feature, meaning they can detect and classify objects in an image at high speed, allowing for the scalability of the developed algorithm. This is a critical feature, meaning they can detect and classify objects in an image at high speed, allowing for the scalability of the developed algorithm.

Figure 2. Synopsis flowchart of the developed treefall algorithm.

3.1.2. DL Training Model

In the developed method, the workflow initiates with tiled orthomosaic images in the visible spectrum. The images in this study were annotated for two categories: treefall and root ball instances. In the segmentation of treefall instances, focus is placed exclusively on the visible portion of the tree trunk. This includes the area from the base of the lowest point of the tree, extending to where a deciduous tree branches out or encompassing the predominant length of conifer trees. It is important to note that while trees comprise branches and roots, these components are not always visible or discernible in the dataset. This lack of visibility is particularly common in densely forested areas, where treetops often overlap, making them unclear. The annotations were performed by one person, and a second person always performed quality control validation [35]. The annotated images train the selected YOLOv8x-seg model [36]. This is the large instance segmentation-specific version of YOLOv8x-seg developed by Ultralytics, which stands for

Regarding the selection of the image sizes, training documentation for YoloV8 indicates that using higher resolution images is beneficial when training with a high count of small objects, such as treefall instances [37]. Therefore, the image size was set to 1024 by 1024 pixels. This resolution offers a balance between higher-quality images and the ability to use larger batch sizes during training. Additionally, this resolution provides a square tile with an approximate area of 20 m, meeting the desired specifications. For consistent inference and detection, input images of the same size are recommended.

The large instance segmentation model, encompassing 71 million parameters, was selected. Despite a slight reduction in detection speed, this model exhibits the highest mean average precision, for instance, segmentation. This is particularly evident in the intersect over union (IoU) metrics at both 50% and 50–90%, as established during the model's development [38]. The main parts of this model are the head, neck, and backbone. The backbone of this model provides five different scaled features labeled as P1–P5. This architecture is based on the coarse-to-fine (C2f) module that enhances information flow [39]. In this step, the Spatial Pyramid Pooling (SPPF) module adapts the special distribution of the features to create output features. The location of the backbone output features is identified in the neck part of the model. This structure enables the model to predict small details in large images, a key feature in identifying fallen trees in the dataset and improves the precision of the model. Ultimately, the detection is completed in the head part of the model. The head of the model contains two different sections; one of these sections identifies the feature class with binary cross-entropy loss (BCE Loss) for object classification, and the other section identifies the exact location of the object with distribution focal loss (DFL) for bounding box regression.

3.1.1. Data Preparation

A total of 5100 instance-level annotated images have been uploaded to an online platform for management, splitting, and augmentation [40]. These images are annotated for both the treefall and root ball instances, meaning each image may include instances from one or both categories. The dataset was divided into three distinct sets: training, validation, and testing, constituting 60% (3017 images), 20% (1005 images), and 20% (1006 images) of the entire dataset, respectively. Data augmentation was performed to add variation to the dataset and mitigate specific biases. The selection of annotated images encompassed diverse topographical regions of a tornado-damaged forested area, as previously detailed in Section 2, to accommodate variations in resolution, lighting, and tree species. This process was carried out to balance the dataset in terms of composition.

Figure 3 presents an illustrative example of annotated images, showing scenarios where trees are only partially captured or completely absent, along with the presence of excessive shadows. These images were left unaltered as such scenarios are expected in the data used by the trained deep learning model. During the annotation process, only significant and visible instances of treefall and root ball were annotated. The aim is to use these annotations for further processing to quantify the treefall angle. In these images, the blue polygons denote fallen tree trunks, while the red polygons represent the root balls. It is noteworthy that the root balls are reserved for future use in the annotated data and the developed model, as they are not explicitly utilized in the current treefall detection algorithm.

Image augmentation is conducted in this study due to the substantial number of learnable parameters in the yolov8x-seg deep learning model. Within the proposed method, augmentation enhances model generalizability by introducing both data variation as well as an increase in the number of training instances. However, it is pivotal to carefully choose an augmentation method that aligns with the potential challenges present in the dataset [41]. In this algorithm development, eight image augmentation techniques, including flip, rotate, shear, crop and rotation, noise, blur, and cutout, were employed to increase variability. Additionally, applying augmentation techniques not only increased the size of the training dataset but also its variation, reducing potential overfitting. During initial training sessions, it was noticed that augmentation boosted the mAP50 value by about 6–8%, which is crucial for the effective deployment of the model. This positive impact of augmentation aligns with findings from previous studies [42]. It is worth noting that the use of noise, blur, and cutouts [43], while not common in augmentation techniques, was intentionally incorporated to address the peculiarities of less-than-ideal orthomosaic images in nature [44]. Mosaiced images are susceptible to blur, ghosting, and other irregularities, and the inclusion of noise, blur, and cutouts contributes to the development of a robust model.

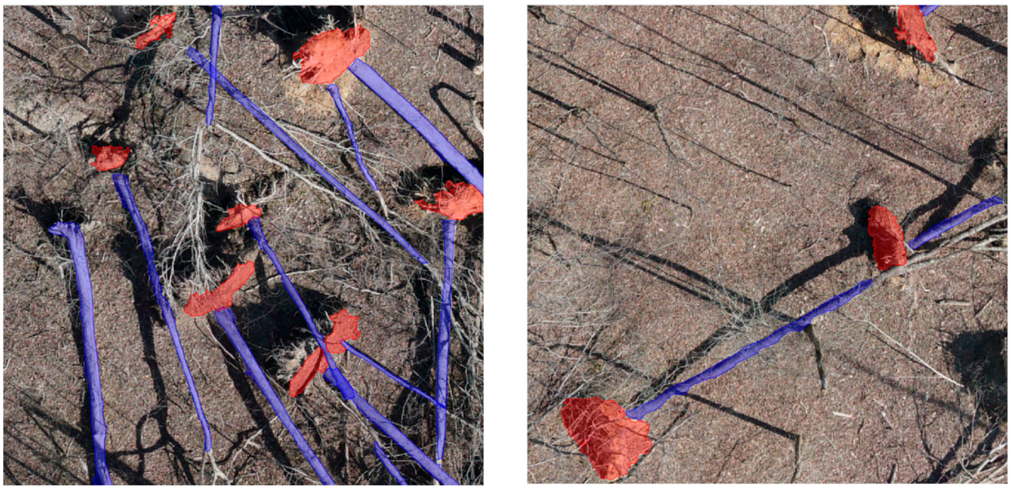
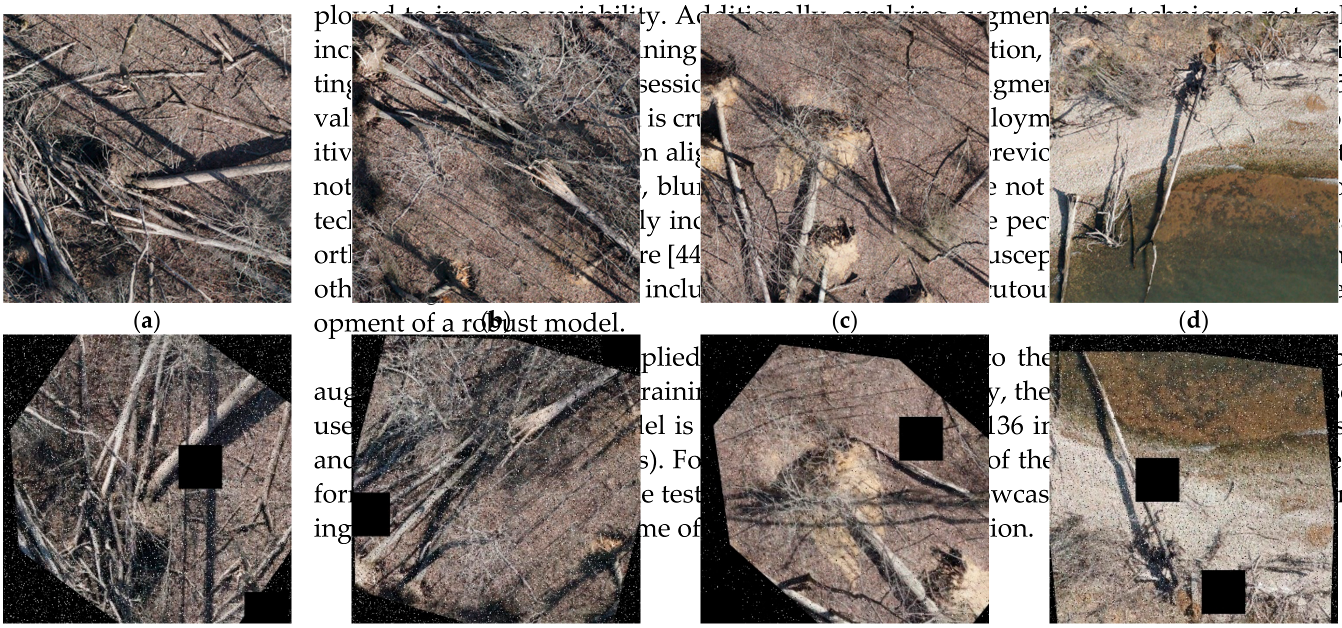


Figure 3. Example input image annotations: (**a**) shows a high density of fallen trees, while (**b**) depicts a low density of fallen trees. In this image, the blue and red colors are used to represent instances of treefall and root ball, respectively.

Figure 3. Example input image annotations: (**a**) shows a high density of fallen trees, while (**b**) depicts a low density of fallen trees. In this image, the blue and red colors are used to represent instances of treefall and root ball, respectively.

Augmentation was applied randomly exclusively to the training images, thereby augmenting the size of the training set only. Consequently, the distribution of the dataset used in the developed model is as follows: training (24,136 images), test (1006 images), and validation (1005 images). Following the completion of the training process, the performance is evaluated on the testing dataset. Figure 4 showcases an example of the training images, highlighting some of the effects of augmentation.

Image augmentation is conducted in this study due to the substantial number of learnable parameters in the yolov8x-seg deep learning model. Within the proposed method, augmentation enhances model generalizability by introducing both data variation as well as an increase in the number of training instances. However, it is pivotal to carefully choose an augmentation method that aligns with the potential challenges present in the dataset [41]. In this algorithm development, eight image augmentation techniques, including flip, rotate, shear, crop and rotation, noise, blur, and cutout, were employed to increase variability. Additionally, applying augmentation techniques not only



opment of a robust model.

Figure 4. Examples of training datasets used for the model: (**a**–**d**) original images and (**e**–**h**) augmented versions of the same images, respectively.

Figure 4. Examples of training datasets used for the model: (**a**–**d**) original images and (**e**–**h**) augmented versions of the same images, respectively.

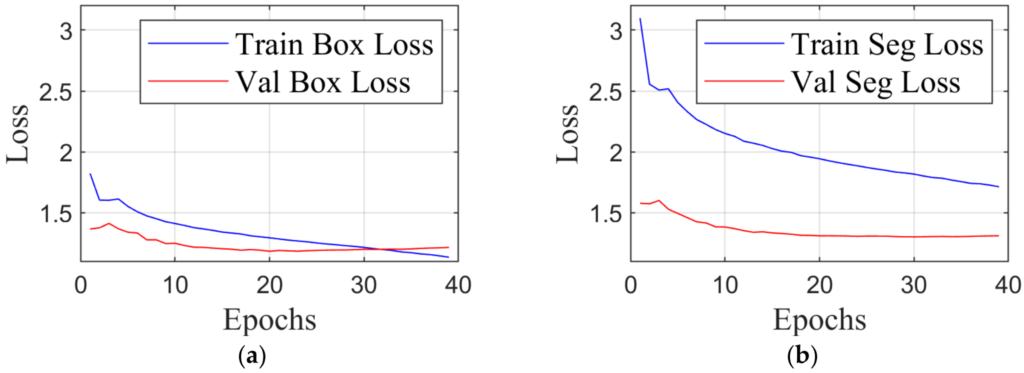
3.1.2. Deep Learning Model Performance

The model's training initially took place online [40], but it was subsequently iterated and refined using local resources, resulting in improved predictions as anticipated. The model was trained in a Linux-based environment, utilizing an Intel Xeon Gold 5218 CPU, 248 GB of RAM, and an NVIDIA RTX™ A6000 graphics processing unit (GPU) equipped with 48 GB GDDR6 memory and 10,752 CUDA cores. This GPU is optimized for artificial

3.1.2. Deep Learning Model Performance

The model's training initially took place online [40], but it was subsequently iterated and refined using local resources, resulting in improved predictions as anticipated. The model was trained in a Linux-based environment, utilizing an Intel Xeon Gold 5218 CPU, 248 GB of RAM, and an NVIDIA RTX™ A6000 graphics processing unit (GPU) equipped with 48 GB GDDR6 memory and 10,752 CUDA cores. This GPU is optimized for artificial intelligence applications.

Based on the results of hyperparameter tuning, the maximum number of epochs considered was set at 40. Figure 5 illustrates the training and validation loss curves over a 40-epoch training. As shown in Figure 5a, a notable observation in the validation errors revealed that the model started to diverge slightly after 20 epochs. Consequently, epoch 20, identified during training, represented the optimal model and was selected for implementation in the algorithm. As depicted in Figure 5, the validation loss is lower than the training loss. While this outcome is not commonly expected, it can be attributed to the extensive data augmentation, incorporating noise, blurring, and cutouts. The results demonstrate that the trained model exhibits reliable predictive performance on the validation dataset.



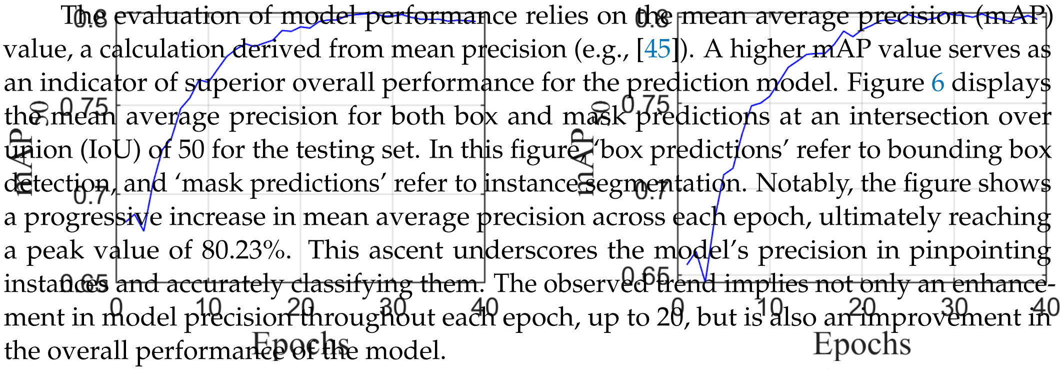
(**a**)

(**b**)

Figure 5. The training and validation loss against the number of epochs for (**a**) object detection and (**b**) instance segmentation of the model.

Figure 5. The training and validation loss against the number of epochs for (**a**) object detection and (**b**) instance segmentation of the model.

The evaluation of model performance relies on the mean average precision (mAP) value, a calculation derived from mean precision (e.g., [45]). A higher mAP value serves as an indicator of superior overall performance for the prediction model. Figure 6 displays the mean average precision for both box and mask predictions at an intersection over union (IoU) of 50 for the testing set. In this figure, 'box predictions' refer to bounding box detection, and 'mask predictions' refer to instance segmentation. Notably, the figure shows a progressive increase in mean average precision across each epoch, ultimately reaching a peak value of 80.23%. This ascent underscores the model's precision in pinpointing instances and accurately classifying them. The observed trend implies not only an enhancement in model precision throughout each epoch, up to 20, but is also an improvement in the overall performance of the model.



(**a**)

(**b**)

Figure 6. Mean average precision curves of the training. (**a**) Curve of box mAP in IoU 50. (**b**) Curve of mask mAP in IoU 50.

As shown in Figure 5, At the onset of training (epoch 1), the loss (error) is lower for object detection, with a value of 1.82, as depicted in Figure 5a, compared to instance segmentation, which shows a loss of 3.09 as illustrated in Figure 5b. This difference can be attributed to the complexity of the segmentation task. However, as shown in Figure 6, the mean precision of the model in both detection and segmentation became equal and reached 80.23% and 80.22%, respectively. An equal mean average precision (mAP) value observed at the end of the training indicates a balanced performance of the model in both localization and object segmentation. Here, localization pertains to accurately identifying the bounding box of the object (treefall or root instance), whereas object segmentation involves delineating the precise outline of the object within the mask prediction. This

As shown in Figure 5, At the onset of training (epoch 1), the loss (error) is lower for object detection, with a value of 1.82, as depicted in Figure 5a, compared to instance segmentation, which shows a loss of 3.09 as illustrated in Figure 5b. This difference can be attributed to the complexity of the segmentation task. However, as shown in Figure 6, the mean precision of the model in both detection and segmentation became equal and reached 80.23% and 80.22%, respectively. An equal mean average precision (mAP) value observed at the end of the training indicates a balanced performance of the model in both localization and object segmentation. Here, localization pertains to accurately identifying the bounding box of the object (treefall or root instance), whereas object segmentation involves delineating the precise outline of the object within the mask prediction. This equality in mAP values suggests that the model is equally proficient in predicting both the bounding box and the mask with similar levels of precision.

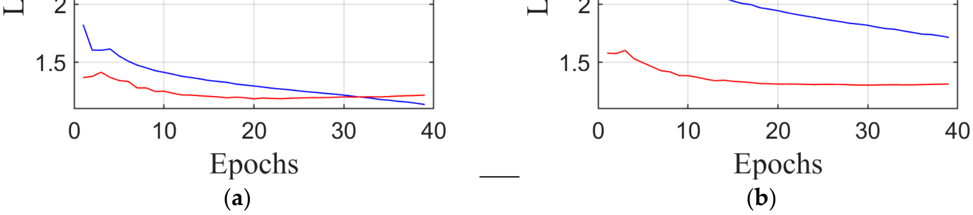


Figure 5. The training and validation loss against the number of epochs for (**a**) object detection and (**b**) instance segmentation of the model.

equality in mAP values suggests that the model is equally proficient in predicting both the bounding box and the mask with similar levels of precision.

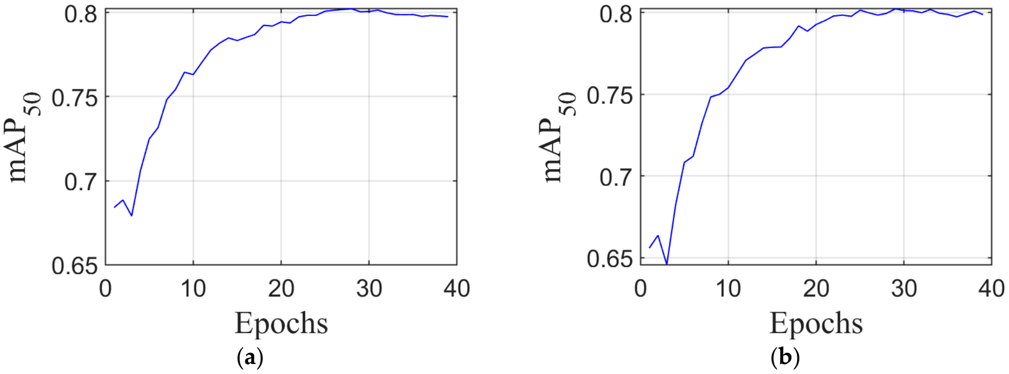


Figure 6. Mean average precision curves of the training. (**a**) Curve of box mAP in IoU 50. (**b**) Curve of mask mAP in IoU 50.

Figure 6. Mean average precision curves of the training. (**a**) Curve of box mAP in IoU 50. (**b**) Curve of mask mAP in IoU 50.

As shown in Figure 5: At the onset of training (epoch 1), the loss (error) is lower for object detection, with a value of 1.82, as depicted in Figure 5a, compared to instance segmentation, which shows a loss of 3.09 as illustrated in Figure 5b. This difference can be attributed to the complexity of the segmentation task. However, as shown in Figure 6, the mean precision of the model in both detection and segmentation became equal and reached 80.23% and 80.22%, respectively. An equal mean average precision (mAP) for the observed at the end of the training indicates a balanced performance of the model in both localization and object segmentation. Here, localization pertains to accurately identifying the bounding box of the object (treefall or root instance), whereas object segmentation involves delineating the precise outline of the object within the mask prediction. This equality in mAP values suggests that the model is equally proficient in predicting both the bounding box and the mask with similar levels of precision.

A normalized confusion matrix was employed, given the imbalance in the number of 'treefall' and 'root' instances in the training dataset. Therefore, considering their proportion is crucial as it offers valuable insights into the classification performance of the deep learning mod1024el on the validation data. The confusion matrix in this study distinguishes among three classes: root (balls), treefall, and background. The background class, a default category in YOLO models, aids in identifying false-positive and false-negative predictions for the annotated classes. It represents areas of the images where the model does not make any predictions. While this class is instrumental in determining the rates of false positives and negatives, it is not independently considered in the calculation of performance metrics [46]. Figure 7 presents the confusion matrix constructed based on the count of each prediction in the validation dataset. In this matrix, a false-negative value represents a missed detection, where a false-negative rate of 0.24 in the background class indicates that 24% of the actual treefall instances were mistakenly considered background, leading to missed detections.

A normalized confusion matrix was employed, given the imbalance in the number of 'treefall' and 'root' instances in the training dataset. Therefore, considering their proportion is crucial as it offers valuable insights into the classification performance of the deep learning mod1024el on the validation data. The confusion matrix in this study distinguishes among three classes: root (balls), treefall, and background. The background class, a default category in YOLO models, aids in identifying false-positive and false-negative predictions for the annotated classes. It represents areas of the images where the model does not make any predictions. While this class is instrumental in determining the rates of false positives and negatives, it is not independently considered in the calculation of performance metrics [46]. Figure 7 presents the confusion matrix constructed based on the count of each prediction in the validation dataset. In this matrix, a false-negative value represents a missed detection, where a false-negative rate of 0.24 in the background class

Additionally, the high false-positive rate of 0.87 is attributed to the frequent misclassification of long fallen tree trunks in numerous instances and some branches as 'treefall'. The notable occurrence of false negatives (0.87) in the background for treefall suggests a high missed detection rate for fallen trees. Nevertheless, the misdetection rate remains relatively low. In these results, many of the undetected trees were slender, resembling branches due to their reduced thickness. While it might be assumed that slender trees have lower wind resistance, they can actually be more resistant due to flexible trunks and smaller surface areas for wind pressure. Despite not providing a representative indication of the prevailing wind direction, these trees were retained in the imagery dataset during preprocessing to avoid introducing bias. Consequently, they may not provide a representative indication of the prevailing wind direction. Moreover, it is anticipated that many instances of treefalls will be averaged to provide a representative wind direction in a given grid dimension; this will be discussed in later sections.

Table 1 presents the precision, recall (sensitivity), F1 score, and specificity calculated for the prediction model according to the following equations:

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \tag{1}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \tag{2}$$

The notable occurrence of false negatives (0.87) in the background for treefall suggests a high missed detection rate for fallen trees. Nevertheless, the misdetection rate remains relatively low. In these results, many of the undetected trees were slender, resembling branches due to their reduced thickness. While it might be assumed that slender trees have lower wind resistance, they can actually be more resistant due to flexible trunks and smaller surface areas for wind pressure. Despite not providing a representative indication of the prevailing wind direction, these trees were retained in the imagery dataset during preprocessing to avoid introducing bias. Consequently, they may not provide a representative indication of the prevailing wind direction. Moreover, it is anticipated that many instances of treefalls will be averaged to provide a representative wind direction in a given grid dimension; this will be discussed in later sections.

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

$$\text{Specificity} = \frac{\text{True Negative}}{\text{True Negative} + \text{False Positive}} \tag{4}$$

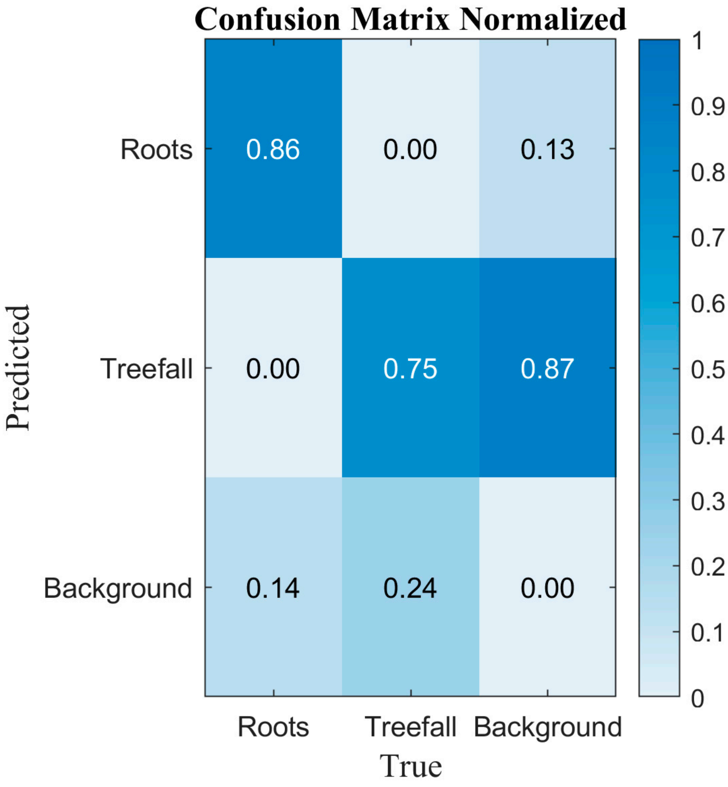


Figure 7. Normalized confusion matrix.

Figure 7. Normalized confusion matrix.

Table 1 presents the precision, recall (sensitivity), F1 score, and specificity calculated for the prediction model according to the following equations:

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \tag{1}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \tag{2}$$

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

$$\text{Specificity} = \frac{\text{True Negative}}{\text{True Negative} + \text{False Positive}} \tag{4}$$

Examining the data in Table 1, it becomes evident that the precision for the root ball class is 85.96%. This figure suggests a high level of accuracy, with the deep learning model correctly identifying root balls in 85.96% of cases. Additionally, the recall (sensitivity) for this class is 79.99%, indicating the model's capability to correctly classify nearly 80% of the root ball instances encountered. Collectively, these metrics result in an F1 score of 0.83, a value typically considered 'good' in practice, especially when it exceeds the threshold of 0.7 [47]. The 'treefall' class exhibits a precision rate of 75.46%, which is slightly lower than that of the 'root' class. However, since this value surpasses 70%, it generally falls within the acceptable range in accordance with common guidelines.

Table 1. Precision, recall (sensitivity), F1 score, and specificity of the YOLOv8x-seg.

Class	Precision (%)	Recall (%)	F1 Score	Specificity
Treefall	75.46	64.02	0.69	0.36
Root	85.96	79.99	0.83	0.92

Furthermore, this performance aligns with the objectives set for the model. Moreover, the F1 score, representing the harmonic mean between precision and recall, supports this conclusion. The F1 score for the 'root' prediction class is 0.83, signifying excellent model performance in predicting roots in images. The F1 score for the 'treefall' class stands at 0.69, which is just shy of the 0.70 benchmark. Despite being slightly below this threshold, this score is deemed acceptable, aligning with the deep learning model's objectives and intended application. In practical deployment, it is expected that averaging multiple instances of treefalls will enhance reliability, particularly in the generation of a wind direction map [47]. Although the 'treefall' class has a lower F1 score due to a higher false-negative rate in predictions, it still falls within a generally acceptable range. A false-negative value signifies a missed detection in each prediction class. In this research, the primary objective is to identify a fallen tree polygon and represent it with a vector arrow. Accurate identification of the fallen tree is crucial, making high precision less critical. This is further emphasized as many fallen trees will be averaged to quantify a representative wind direction in a specific grid size, as will be discussed.

3.2. *Taper Estimation*

The proposed method leverages image processing techniques to process the information extracted from the deep learning model. This approach utilizes the predicted annotation point coordinates, extracted from model-predicted polygons in the JSON files, to estimate tree taper. The presumption is made that the tree stem diameter follows a linear decrease with tree height, where the approximate slope of the tree diameter represents the average tree taper rate [48]. Consequently, the coordinates of treefall pixel points are extracted to quantify tree taper and, consequently, determine the direction of the tree. Figure 8 illustrates an example where one of the tree outlines is imported for analysis in the local image coordinates (in pixels).

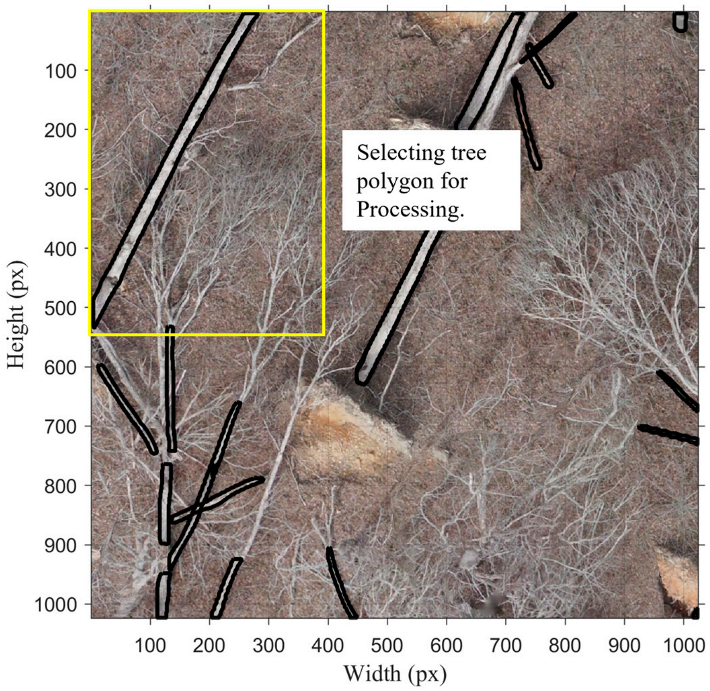


Figure 8. The boundary of the tree trunk is exported into the image coordinate.

Figure 8. The boundary of the tree trunk is exported into the image coordinate.

With the imported tree outlines, the initial step involves determining the midpoint of the trunk as a function of height. Acknowledging that most trees exhibit linear characteristics, but some may be curved, especially after windstorms, a third-degree nonlinear regression is employed to approximate the middle section of the trunk. Utilizing this curve, the two edges of the tree trunk are segmented into positive and negative components (Figure 9), and the distance between them is subsequently measured to quantify the tree taper rate or slope. Figure 9c demonstrates an illustrative taper rate.

With the imported tree outlines, the initial step involves determining the midpoint of the trunk as a function of height. Acknowledging that most trees exhibit linear characteristics, but some may be curved, especially after windstorms, a third-degree nonlinear regression is employed to approximate the middle section of the trunk. Utilizing this curve, the two edges of the tree trunk are segmented into positive and negative components (Figure 9), and the distance between them is subsequently measured to quantify the tree taper rate or slope. Figure 9c demonstrates an illustrative taper rate.

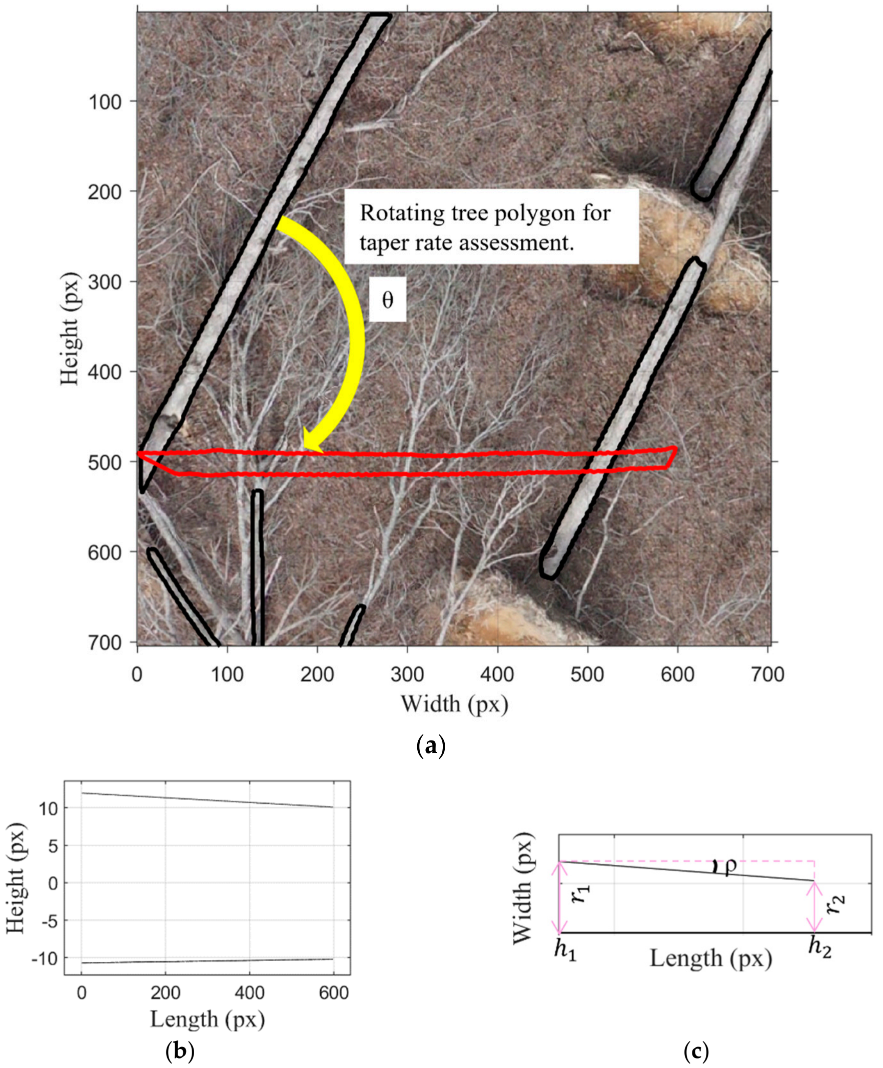


Figure 9. Treefall trunk polygon example: (**a**) rotating tree polygon for taper assessment; (**b**) linear edge lines replaced with the boundary of the tree polygon; (**c**) illustration of tree taper rate, denoted as ρ.

With a linear taper rate and the variables shown in Figure 9, the taper rate is computed as shown in Equation (5):

$$\rho = \frac{r_2 - r_1}{h_2 - h_1} \quad (5)$$

where r_1 and r_2 represent the widths of the tree trunk and $h_2 - h_1$ is the length of the tree polygon. To assess the validity of the calculations of the taper rate, the computed values are compared with the recorded taper rate values in the previous studies [48]. It was found that the remotely sensed tree taper values are in close agreement with their field measurements. Subsequently, the direction of the fallen trees is assigned based on the sign of the taper rate (increasing or decreasing). In this method, the top of the tree is identified in the direction where the taper rate is negative, indicating a decrease in the trunk's diameter with an increase in the height of the standing tree [48]. Figure 10 shows the predicted fall direction for the tree polygon. Note this predicted fall direction is in agreement with the yellow-boxed polygon in Figure 8. This step completes the basic treefall detection and directional assignment; however, additional steps are needed to keep only statistically confident detections and convert them to real-world units.

Figure 9. Treefall trunk polygon example: (**a**) rotating tree polygon for taper assessment; (**b**) linear edge lines replaced with the boundary of the tree polygon; (**c**) illustration of tree taper rate, denoted as ρ.

With a linear taper rate and the variables shown in Figure 9, the taper rate is computed as shown in Equation (5):

$$\rho = \frac{r_2 - r_1}{h_2 - h_1} \quad (5)$$

where r_1 and r_2 represent the widths of the tree trunk and $h_2 - h_1$ is the length of the tree polygon. To assess the validity of the calculations of the taper rate, the computed values are compared with the recorded taper rate values in the previous studies [48]. It was found that the remotely sensed tree taper values are in close agreement with their field measurements. Subsequently, the direction of the fallen trees is assigned based on the sign of the taper rate (increasing or decreasing). In this method, the top of the tree is identified in the direction where the taper rate is negative, indicating a decrease in the trunk's diameter with an increase in the height of the standing tree [48]. Figure 10 shows the predicted fall direction for the tree polygon. Note this predicted fall direction is in agreement with the yellow-boxed polygon in Figure 8. This step completes the basic treefall detection and directional assignment; however, additional steps are needed to keep only statistically confident detections and convert them to real-world units.

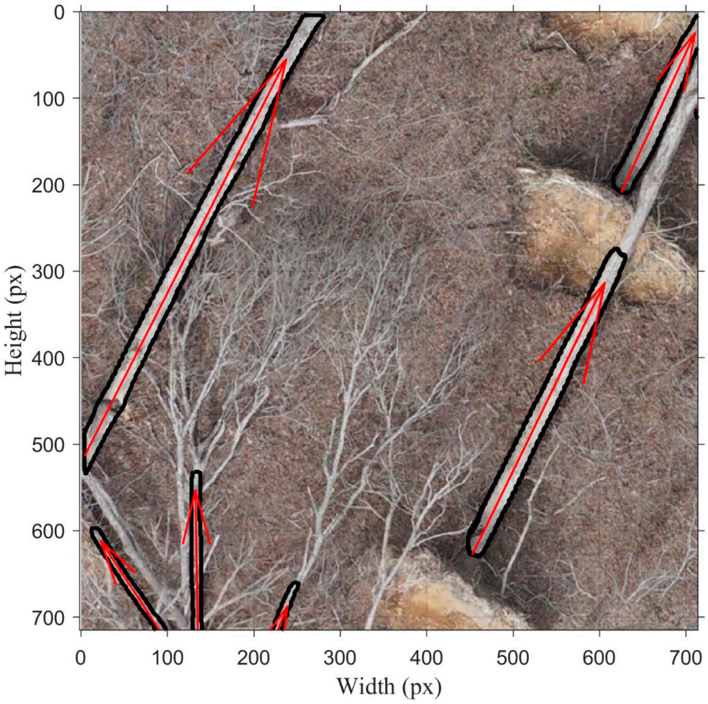


Figure 10. Estimated direction for one example tree polygon. The black polygon represents the boundary of the tree trunk predicted by the model, and the red arrow represents the estimated direction of the fallen tree.

3.3.3. Parametric Analysis

Different statistical techniques were employed to reduce mispredictions, which may occur due to discrepancies in the tree polygon resulting from pixel distortion or errors in the instance segmentation model. Here, the primary aim of the parametric analysis is to remove unreliable assigned directions or mispredictions. This is performed by identifying and filtering out distorted tree polygons. This is performed in an empirical manner, but this also instills some interpretability in the detected model as it provides some physical and realistic bounds on the detections based on known characteristics of the fallen trees and the imagery dataset.

In this analysis, statistical parameters employed for filtering encompass (1) taper rate, indicating the change in tree diameter over a specific tree length; (2) confidence level, representing the probability that the prediction is correct; (3) thickness range; and (4) the coefficient of determination. The first parameter of the tree taper is established to remove polygons that do not have a pronounced taper. Secondly, the confidence level relates to the model's detection confidence. The thickness range removes small-diameter trees, branches, and other features that are not as reliable and/or characterized by lower wind resistance. The last filter parameter is also a very efficient geometric descriptor, the coefficient of determination (R2), which evaluates how well a linear model fits the dataset. This coefficient value ranges from 0 to 1, with higher values indicating a better fit of the linear model to the data.

A parametric analysis was performed to achieve the best value for each of the statistical parameters. Here, the goal was to define a threshold value that increases the efficiency of the predicted directions by removing the inaccurate directions and retaining as many correct directions as possible. To this end, a randomly identified subset of 7275 images was selected, totaling 78,059 instances. These instances provide a comprehensive representation of the dataset, encompassing various tree species, thicknesses, heights, and environmental effects, including shadows. Different threshold values were independently applied to various parameters. The impact of these thresholds on the accuracy of the remaining dataset was then assessed and documented, focusing on the number of mispredictions and

Figure 10. Estimated direction for one example tree polygon. The black polygon represents the boundary of the tree trunk predicted by the model, and the red arrow represents the estimated direction of the fallen tree.

3.3.3. Parametric Analysis

Different statistical techniques were employed to reduce mispredictions, which may occur due to discrepancies in the tree polygon resulting from pixel distortion or errors in the instance segmentation model. Here, the primary aim of the parametric analysis is to remove unreliable assigned directions or mispredictions. This is performed by identifying and filtering out distorted tree polygons. This is performed in an empirical manner, but this also instills some interpretability in the detected model as it provides some physical and realistic bounds on the detections based on known characteristics of the fallen trees and the imagery dataset.

In this analysis, statistical parameters employed for filtering encompass (1) taper rate, indicating the change in tree diameter over a specific tree length; (2) confidence level, representing the probability that the prediction is correct; (3) thickness range; and (4) the coefficient of determination. The first parameter of the tree taper is established to remove polygons that do not have a pronounced taper. Secondly, the confidence level relates to the model's detection confidence. The thickness range removes small-diameter trees, branches, and other features that are not as reliable and/or characterized by lower wind resistance. The last filter parameter is also a very efficient geometric descriptor, the coefficient of determination (R2), which evaluates how well a linear model fits the dataset. This coefficient value ranges from 0 to 1, with higher values indicating a better fit of the linear model to the data.

A parametric analysis was performed to achieve the best value for each of the statistical parameters. Here, the goal was to define a threshold value that increases the efficiency of the predicted directions by removing the inaccurate directions and retaining as many correct directions as possible. To this end, a randomly identified subset of 7275 images was selected, totaling 78,059 instances. These instances provide a comprehensive representation of the dataset, encompassing various tree species, thicknesses, heights, and environmental effects, including shadows. Different threshold values were independently

the accuracy of treefall angle predictions. The optimal threshold for each parameter was selected to strike a balance between significantly reducing mispredictions and maintaining accurate predictions. Table 2 outlines these specific threshold values. By applying these thresholds, it was possible to eliminate 34% of the inaccurate predictions from the dataset. It is important to note that the range of these threshold values might vary in other datasets due to differences in ground sampling distance (GSD) and data quality (e.g., blurriness). Therefore, it is expected that each statistical parameter discussed in this study will be fine-tuned when evaluating different tornado tracks for optimal effectiveness. Table 2 shows the optimized threshold value determined based on the parametric evaluations in this research. As a result of these filters, all of the tree trunks that are extremely curved, have flat edges, or have unusual taper rates are removed. It is also noted that while these parameters are determined empirically on a subset of the data, they provide a physical context to the detections, increasing the interpretability of the deep learning model and the algorithm that was developed.

Table 2. Optimized threshold values for filtering.

Parameters	Values
Thickness range	<1.2
Taper rate	<0.002
Confidence level	<0.2
Coefficient of determination	<0.01

3.4. Geolocating the Results and Averaged Representative Directions

In the algorithm to date, the predicted treefall directions are located in the local image coordinate system for each instance. For a comprehensive assessment of real tornadoes, it is essential to scale and translate the results into real-world coordinates. This requires a conversion of the pixel coordinates to projected coordinates. Projected coordinates are initially selected in lieu of a geodetic coordinate system as they relate to localization in units of length, with meters being the most common, which is critical for the last step of the algorithm. The imported file types are GeoTIFFs, where the coordinate system of the image is defined and can be applied on a per-image basis to the treefall instances. This is performed for all images that have detected treefall instances and met the filtering criteria, and the pixel coordinates are converted to projected coordinates (in length units). In a final and exported step, the start location and end location of each of the detected treefalls are also translated to WGS-84 values of latitude and longitude along with the predicted fall direction (azimuth angle) for ease in geographical information system software (GIS) platforms.

Due to the large count of the detected fallen trees and their predicted fall angles, this can be complex and cumbersome in data size. Consequently, a box method is employed to determine the representative fall direction in a determined grid cell size. To enhance the efficiency of the assessment, each box is evaluated separately [49]. This step is commonly performed in the treefall methods as it provides a more reliable assessment of the wind direction on average. In this final step, an additional requirement of a minimum number of fallen trees is established to account for areas where the treefall is sparse. For the selected grid size, the median direction of fallen trees is calculated for grid cells exceeding the minimum count of fallen trees. The grid size is fixed at 75 m by 75 m, taking into account the width of the tornado damage path and the density of treefalls, with a threshold of at least 10 trees per cell. Although these figures aim to illustrate general trends for this dataset, they are anticipated to be adjusted for other datasets, depending on the particular spatial distribution and density of treefalls in the damage scene. The results of the gridded representative wind directions are illustrated in Figure 11. It is noted that these numbers work well for this dataset, but it is anticipated that the grid size and the minimum number of trees will be dependent on the intensity of the tornado, the width and path of the tornado, and the forest stand structure.

that these numbers work well for this dataset, but it is anticipated that the grid size and the minimum number of trees will be dependent on the intensity of the tornado, the width and path of the tornado, and the forest stand structure.

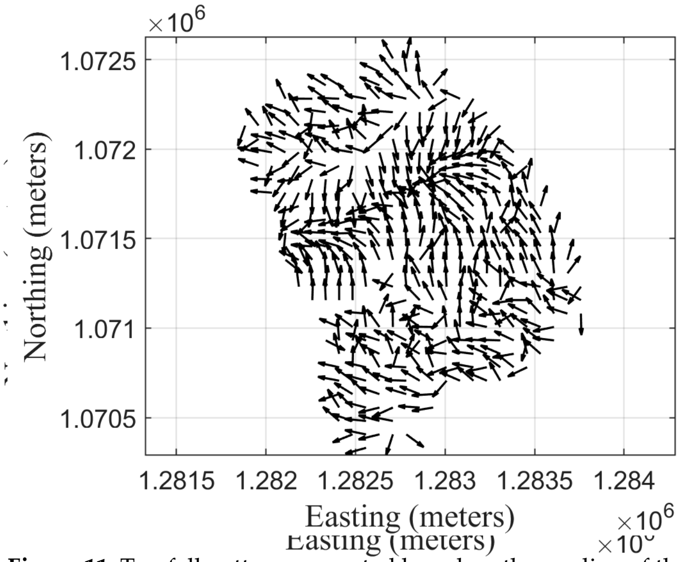


Figure 11. Treefall pattern generated based on the median of the estimated angles.

4. Results and Discussion

To evaluate the model's performance, a study area was selected from two tornado tracks in the Land Between the Lakes National Recreation Area (LBL) in the United States, which was hit by a tornado on 10 December 2021. In this research, the north track was chosen for the model assessment. The northern track is located in Kentucky, and according to the National Weather Service, the tornado was rated as EF-4 [50]. The north track covers a region of approximately 26 km^2, created from the ten surveying zones conducted after the tornado during the research project. The observed damage in the area was mainly large, uprooted trees, but snapped trees and thrown trees were also present. Figure 12 shows the detailed location of the selected study region in red color in Kentucky.

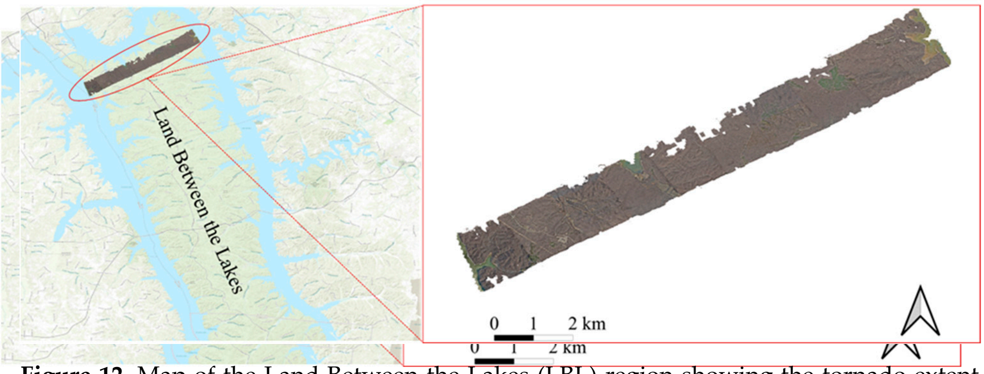


Figure 12. Map of the Land Between the Lakes (LBL) region showing the tornado extent and the location of the surveying sites in red.

Tiled orthophotos, exported as GeoTIFFs, were leveraged and imported into the developed algorithm to detect fallen trees. It is noted that the dataset used to train the model was selected from various surveying zones of this similar region. However, this analysis utilizes the entire track that bisected this section of the forest. In this application, the entire northern track consists of 76,885 orthophotos, which have a ground sampling distance of 2 cm (or better).

After each of these images was run through the trained deep learning model, 907,863 treefall instances were detected. A JSON was exported for each of the images that contain a treefall instance, and a polygon outline was provided for each treefall. Figure 13 demonstrates some detection results on the previously unseen images. This figure shows that the model could predict fallen tree trunks with high confidence levels.

After each of these images was run through the trained deep learning model, 907,863 treefall instances were detected. A JSON was exported for each of the images that contain a treefall instance, and a polygon outline was provided for each treefall. Figure 13 demonstrates some detection results on the previously unseen images. This figure shows that the model could predict fallen tree trunks with high confidence levels. More importantly, the observation supports the high false-negative rate in the confusion matrix. These missed detections correlated with the narrow trees and also trees located in regions with high branches.

More importantly, the observation supports the high false-negative rate in the confusion matrix. These missed detections correlated with the narrow trees and also trees located in regions with high branches.

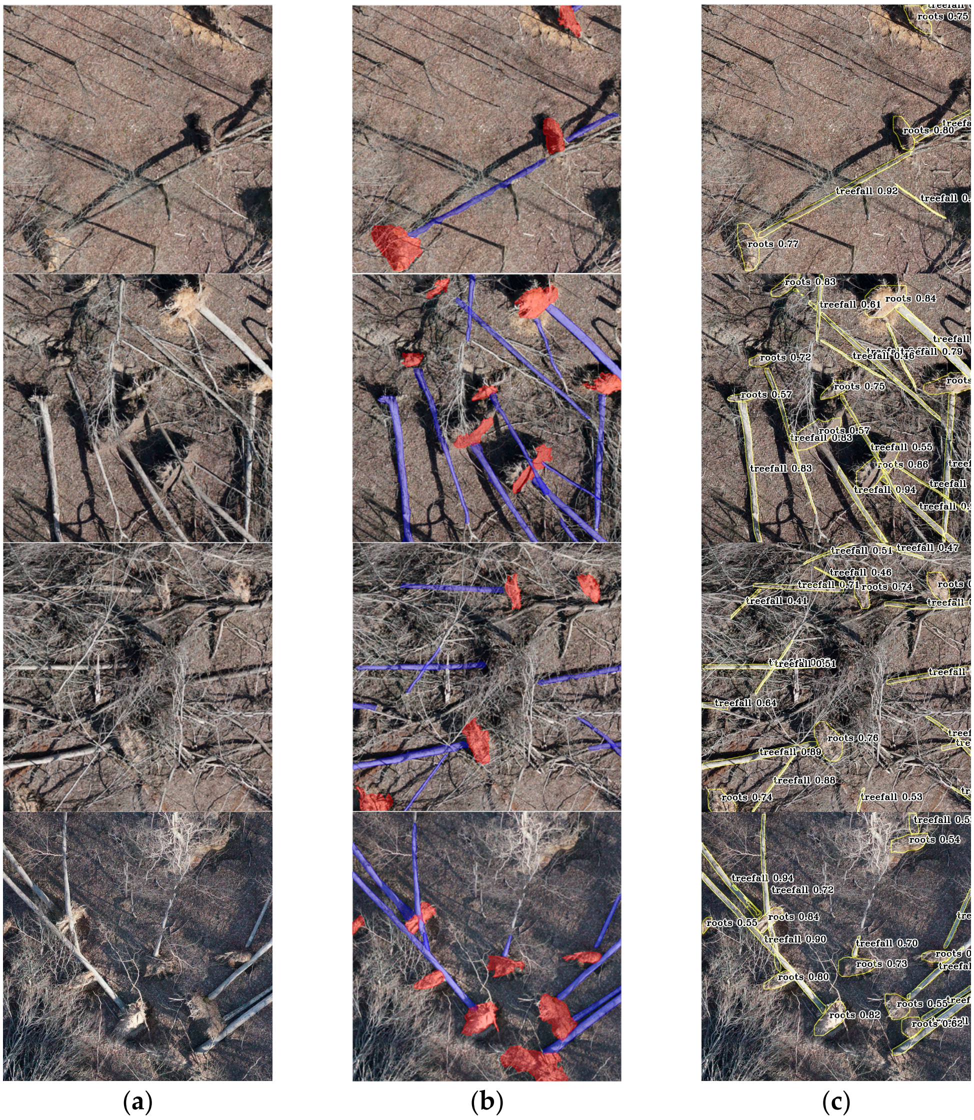


Figure 13. Example input images with annotations: (**a**) original images; (**b**) images with annotations; (**c**) detected instances from the deep learning model. In this figure, blue represents a fallen tree trunk, and red represents a root ball.

Figure 13. Example input images with annotations: (**a**) original images; (**b**) images with annotations; (**c**) detected instances from the deep learning model. In this figure, blue represents a fallen tree trunk, and red represents a root ball.

4.1. Tree Taper and Parametric Filtering

4.1. Tree Taper and Parametric Filtering

In the next step, the fallen tree polygon coordinates are used to quantify the taper rate and its corresponding sign (positive or negative). With the sign of the taper rate, the direction of the fallen tree is identified for each treefall instance. As highlighted in Figure 14, some inaccuracies persist in the direction of the fallen trees. This is a result of some trees being partially covered with other fallen trees due to the dense forest stand; consequently, the model predicts a distorted polygon. In this figure, red indicates the direction of one fallen tree instance.

In the next step, the fallen tree polygon coordinates are used to quantify the taper rate and its corresponding sign (positive or negative). With the sign of the taper rate, the direction of the fallen tree is identified for each treefall instance. As highlighted in Figure 14, some inaccuracies persist in the direction of the fallen trees. This is a result of some trees being partially covered with other fallen trees due to the dense forest stand; consequently, the model predicts a distorted polygon. In this figure, red indicates the direction of one fallen tree instance.

Afterward, the predefined threshold values for the parameters were applied to eliminate perceived unreliable predicted directions. Here, the mispredicted angles have a 180-degree error, which is associated with the distortion in the tree polygon and the unreliability of the tree taper. As outlined earlier, a parametric analysis quantified values that can be used for the taper rate, thickness, confidence level, and coefficient determination

Afterward, the predefined threshold values for the parameters were applied to eliminate perceived unreliable predicted directions. Here, the mispredicted angles have a 180-degree error, which is associated with the distortion in the tree polygon and the unreliability of the tree taper. As outlined earlier, a parametric analysis quantified values that can be used for the taper rate, thickness, confidence level, and coefficient determination (Table 2). This allows for the identification and subsequent removal of distorted annotations. After the filtering operation, 788,332 instances remained, and 119,531 (13%) were removed. This shows that the predictions are mainly logical and accurate.

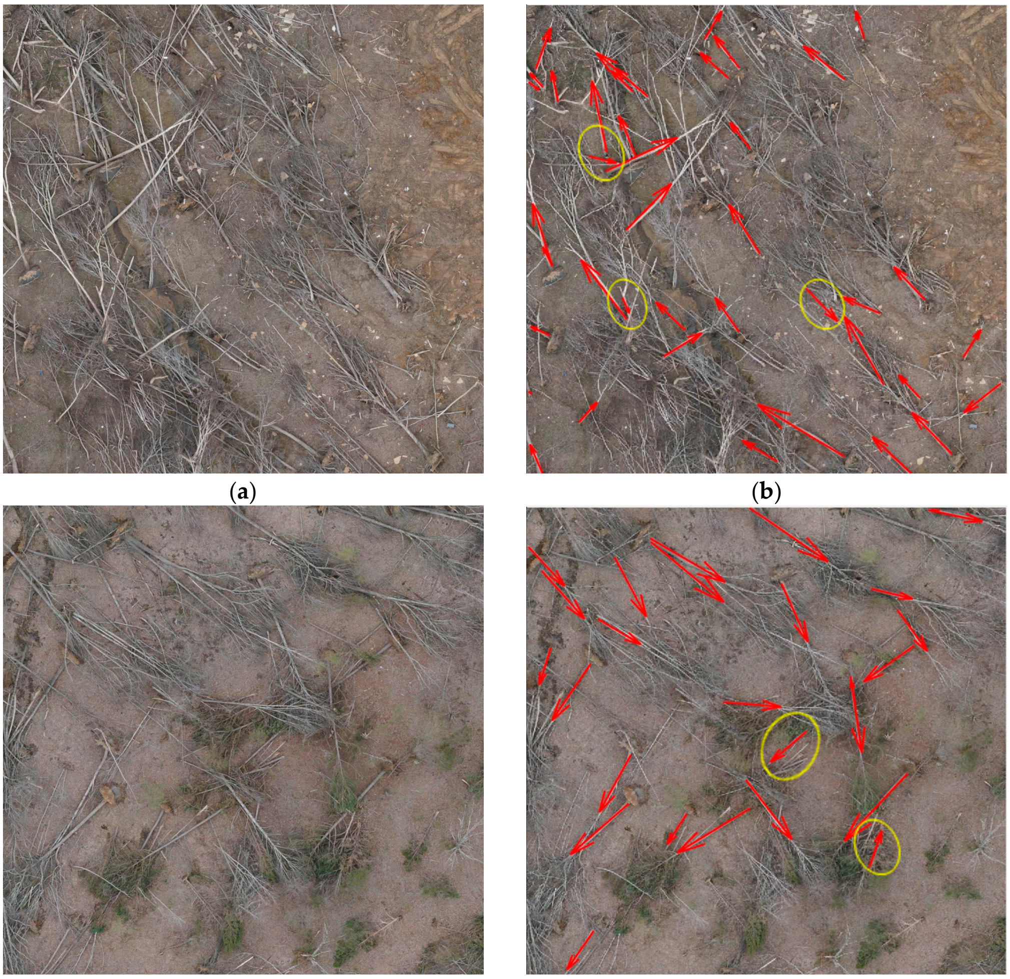
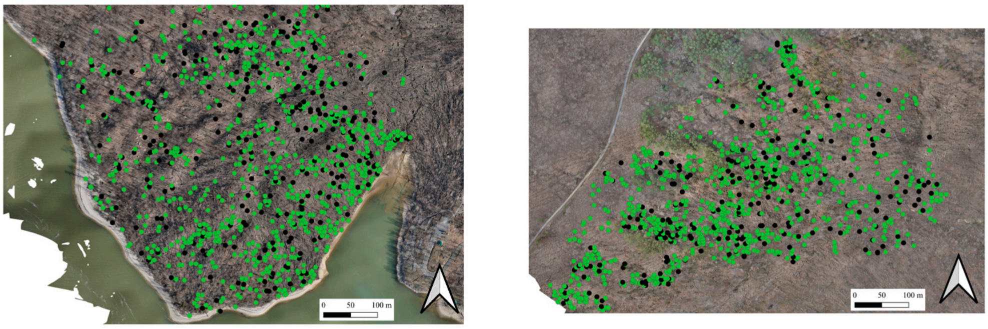

(**a**) (**b**)
(**c**) (**d**)

Figure 14. Estimated treefall direction determined: (**a**,**c**) original image; (**b**,**d**) predicted fall angle overlaid on images. In this figure, the red arrows represent the estimated treefall directions and the yellow circles indicate the inaccurate angle estimations.

4.2. Evaluating Predicted Treefall Direction

To rigorously test the algorithm's ability to accurately predict the direction of treefall using tree taper, two distinct regions, each approximately 500 × 500 m in size, have been selected. These regions were chosen to represent the diversity of the damaged forested area: one region predominantly features deciduous trees without leaves, and the other comprises a mix of deciduous and coniferous trees. Figure 15 visually represents these regions, with the first region, shown in Figure 15a, notably containing many shadows. Across these regions, over 2300 treefall directions were manually annotated to compare with the deep learning model's predicted instances and assigned directions. This process was conducted randomly in these areas, aiming to measure the accuracy of the predicted treefall directions. The analysis, performed on a per-tree instance basis, reveals a weighted average accuracy rate of 73.22% in correctly predicting treefall direction. This metric is derived from 893 correctly predicted directions out of 1238 treefall instances in the first region and 841 correct predictions out of 1130 instances in the second region.

weighted average accuracy rate of 73.22% in correctly predicting treefall direction. This metric is derived from 893 correctly predicted directions out of 1238 treefall instances in the first region and 841 correct predictions out of 1130 instances in the second region.



(**a**) (**b**)

Figure 15. Color-coded results of treefall direction accuracy, with green indicating accurate values and black representing incorrect values, depicted over two zones: (**a**) the first region predominantly populated by deciduous trees with shadows, and (**b**) the second region characterized by a mixture of coniferous and deciduous trees.

Figure 15. Color-coded results of treefall direction accuracy, with green indicating accurate values and black representing incorrect values, depicted over two zones: (**a**) the first region predominantly populated by deciduous trees with shadows, and (**b**) the second region characterized by a mixture of coniferous and deciduous trees.

4.3. Treefall Pattern

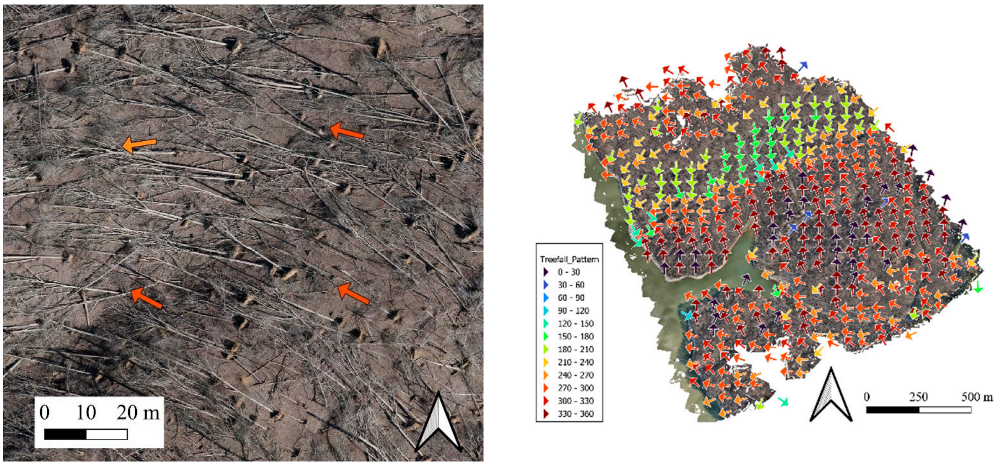
4.3. Treefall Pattern

The directional information from the filtered treefall instances is employed to comprehend the treefall pattern and wind dynamics within the northern track area. As depicted in Figure 16a, each arrow signifies the median of all tree directions established for the northern track for a grid spacing of 75 m. This is expanded to show a larger section in Figure 16b. Moreover, approximately one-third of the northern track is shown in Figure 16c, demonstrating the algorithm's efficiency and scalability. This figure also illustrates the treefall pattern. Concurrently, it reveals a convergent pattern and a broader damaged region on the south side of the path, consistent with prior findings [11].

The digital elevation map was derived by the project team using a publicly available lidar dataset captured in [5], which predates the tornado event. The processed DEM was constructed with a resolution of 2 m. This involved utilizing a digital elevation model (DEM) representing only ground points, colorized by elevation, as depicted in Figure 17. Based on a preliminary assessment of wind direction versus elevation, it appears that at least two interactions may be present in certain regions of the area. The first observed interaction is the tornado deflecting to the left while climbing the windward side of a hill, then deflecting to the right while descending the leeward side [9].

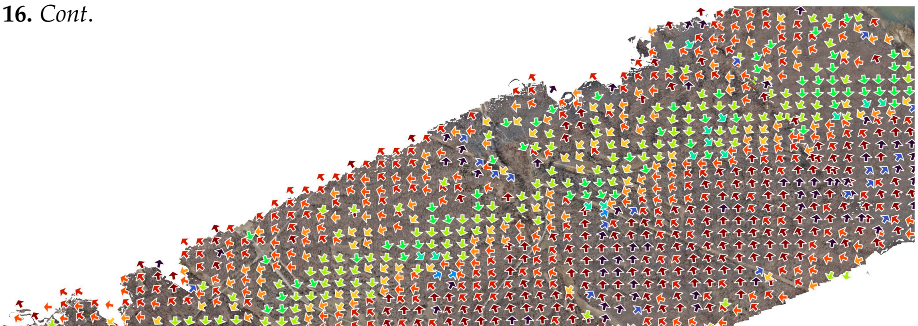
The directional information from the filtered treefall instances is employed to comprehend the treefall pattern and wind dynamics within the northern track area. As depicted in Figure 16a, each arrow signifies the median of all tree directions established for the northern track for a grid spacing of 75 m. This is expanded to show a larger section in Figure 16b. Moreover, approximately one-third of the northern track is shown in Figure 16c, demonstrating the algorithm's efficiency and scalability. This figure also illustrates the treefall pattern. Concurrently, it reveals a convergent pattern and a broader damaged region on the south side of the path, consistent with prior findings [11].

The digital elevation map was derived by the project team using a publicly available lidar dataset captured in [5], which predates the tornado event. The processed DEM was constructed with a resolution of 2 m. This involved utilizing a digital elevation model (DEM) representing only ground points, colorized by elevation, as depicted in Figure 17. Based on a preliminary assessment of wind direction versus elevation, it appears that at least two interactions may be present in certain regions of the area. The first observed interaction is the tornado deflecting to the left while climbing the windward side of a hill, then deflecting to the right while descending the leeward side [9].



(**a**) (**b**)

Figure 16. *Cont.*



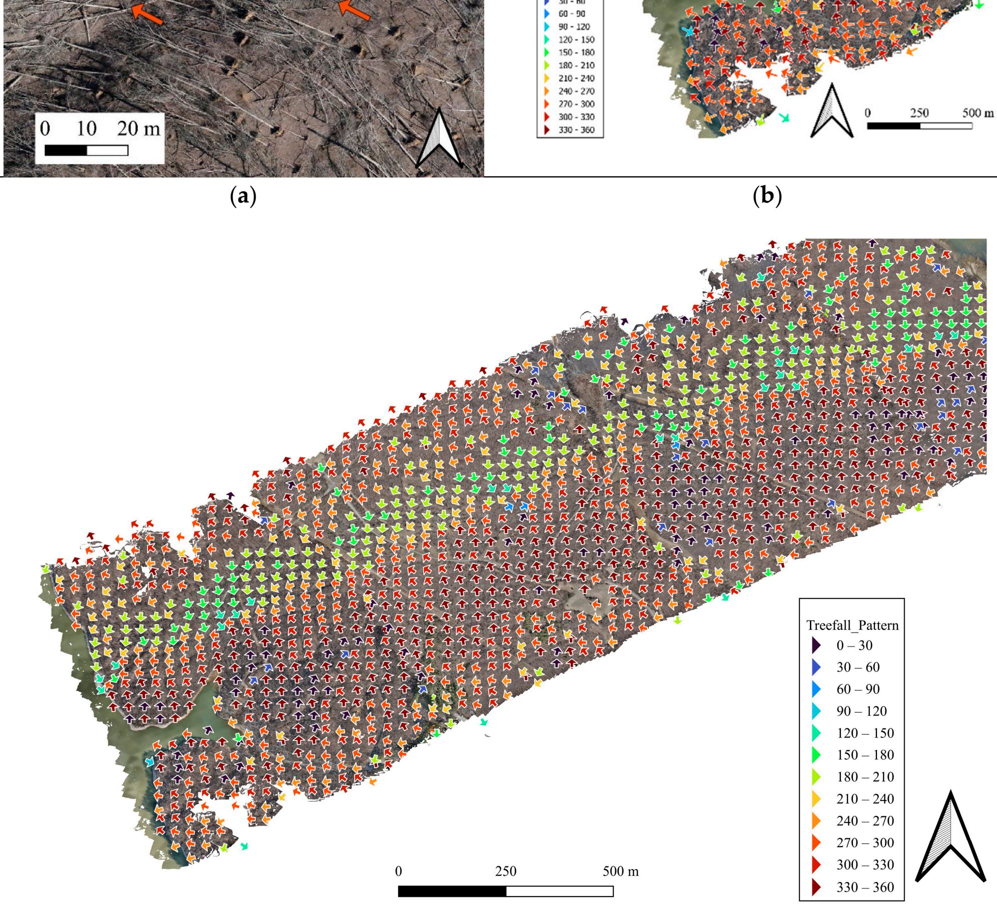


Figure 16. Subsampled wind direction map. Each arrow represents the center of a 75 by 75 m grid: (**a**) detailed view of each grid cell; (**b**) generated results for one survey zone; (**c**) alignment of wind direction maps for different survey zones.

Figure 16. Subsampled wind direction map. Each arrow represents the center of a 75 by 75 m grid: (a) detailed view of each grid cell; (b) generated results for one survey zone; (c) alignment of wind direction maps for different survey zones.

ɔ travel
ons are
mation

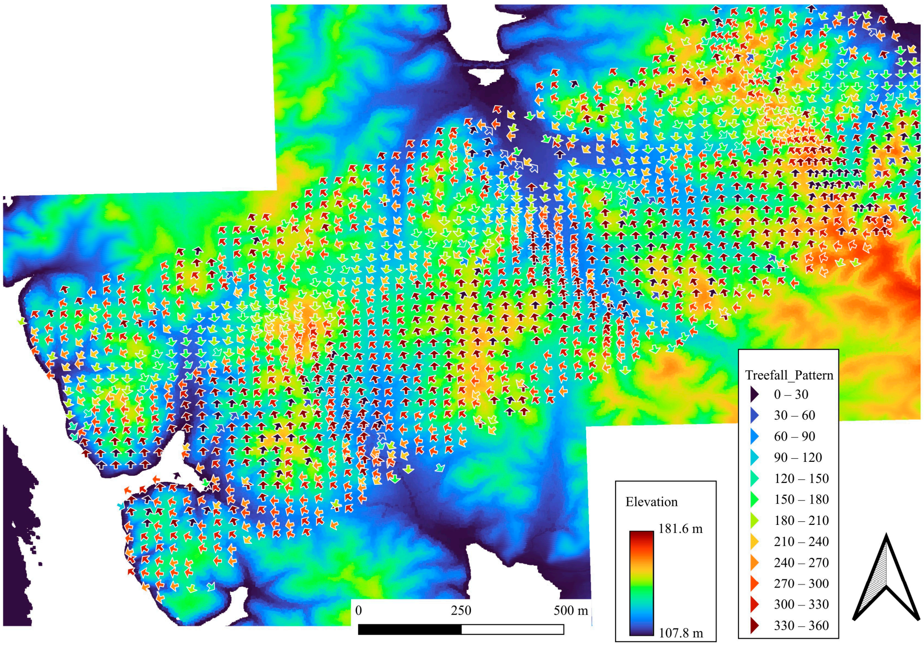


Figure 17. Subsampled colorized wind direction map overlain with the elevation map. Each arrow represents the center of a 75 m by 75 m grid.

Figure 17. Subsampled colorized wind direction map overlain with the elevation map. Each arrow represents the center of a 75 m by 75 m grid.

Moreover, further down the track on the east end, the tornado's path seems to travel along an undulating valley bottom [52]. It is important to note that these observations are preliminary in nature. Further exploration will employ a more robust treefall estimation method [12], directly leveraging the output from the developed algorithm.

5. Conclusions and Future Work

This manuscript introduces and assesses a methodology for detecting fallen trees in tornado-impacted areas using 2D aerial imagery data, eliminating the need for on-site surveying and minimizing access requirements to the affected forested region. The developed method employs YOLOv8x-seg, incorporating object detection and instance segmentation, to identify fallen trees and determine the point coordinates of the trunk polygon for each treefall. Additionally, a tree taper is employed to ascertain the direction of each fallen tree, providing interpretability and filtering distorted polygons when the tree taper is indiscernible.

To assess the method, a dataset from the 2021 Kentucky tornado outbreak is analyzed, relying solely on tiled 2D orthoimages in GeoTIFF format. Due to the extensive vol-

5. Conclusions and Future Work

This manuscript introduces and assesses a methodology for detecting fallen trees in tornado-impacted areas using 2D aerial imagery data, eliminating the need for on-site surveying and minimizing access requirements to the affected forested region. The developed method employs YOLOv8x-seg, incorporating object detection and instance segmentation, to identify fallen trees and determine the point coordinates of the trunk polygon for each treefall. Additionally, a tree taper is employed to ascertain the direction of each fallen tree, providing interpretability and filtering distorted polygons when the tree taper is indiscernible.

To assess the method, a dataset from the 2021 Kentucky tornado outbreak is analyzed, relying solely on tiled 2D orthoimages in GeoTiff format. Due to the extensive volume of fallen trees in the case study, results are subsampled to quantify the representative fall directions in terms of the median direction. The developed method achieves a fallen tree detection accuracy, recall, and precision equal to 74.6%, 64.02%, and 75.46%, respectively. The analysis demonstrates that the method can successfully identify the direction of treefall with an accuracy exceeding 73%. While the proposed method precisely detects the direction for individual trees in unseen data, instances of misalignment by 180 degrees are observed, often attributed to multiple trees falling in the same region and discrepancies in predicted trunk polygons.

The algorithm developed primarily for analyzing tornadoes also holds potential for application in various other types of windstorms. This adaptability broadens the scope of the algorithm, making it a versatile tool in the study of windstorm impacts.

The current model is deliberately tailored for forested regions, excluding suburban and crop areas [53]. Ongoing efforts involve implementing a comparable algorithm with a different deep-learning model designed for these specific scenarios. Regardless of the training dataset's size, all imagery datasets inherently exhibit limited representation in terms of tree species influenced by geographical location. Concurrently, the proposed method has been utilized to evaluate treefall damage following various windstorms. While the results are promising, the necessity for further studies is underscored by the limited size of the data. Such extended research is essential to validate and enhance the reliability of the findings. The continuous enhancement of the dataset's diversity and quantity of tree species remains a critical undertaking poised to have a substantial impact on ongoing model performance refinement.

Author Contributions: Conceptualization, R.L.W.; Methodology, M.N. and R.L.W.; Software, M.N. and R.L.W.; Validation, M.N.; Formal analysis, M.N.; Investigation, M.N. and R.L.W.; Resources, R.L.W.; Data curation, R.L.W.; Writing—original draft, M.N. and R.L.W.; Writing—review and editing, R.L.W.; Supervision, R.L.W.; Project administration, R.L.W.; Funding acquisition, R.L.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Science Foundation (AGS-2221974 and EEC-1950597) and the Northern Tornadoes Project.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Acknowledgments: The authors would like to thank Chris Peterson from the University of Georgia for his feedback regarding tree taper. Finally, the authors acknowledge Cory Lhotka, a graduate student at the University of Illinois at Urbana-Champaign, for his assistance in field data collection. Additionally, Research Experience for Undergraduate (REU) students Chinmoy Dev and Jackalyn H. Wyrobek are greatly appreciated for their help in data annotation.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. November 2023 Tornadoes Report | National Centers for Environmental Information (NCEI). Available online: https://www.ncei.noaa.gov/access/monitoring/monthly-report/tornadoes/202213 (accessed on 26 January 2024).
2. U.S. Economic Damage Caused by Tornadoes 2022 | Statista. Available online: https://www.statista.com/statistics/237409/economic-damage-caused-by-tornadoes-in-us/ (accessed on 26 January 2024).
3. van de Lindt, J.W.; Pei, S.; Dao, T.; Graettinger, A.; Prevatt, D.O.; Gupta, R.; Coulbourne, W. Dual-Objective-Based Tornado Design Philosophy. *J. Struct. Eng.* **2013**, *139*, 251–263. [CrossRef]
4. Wang, Y.; Wang, N.; Lin, P.; Ellingwood, B.; Mahmoud, H.; Maloney, T. De-aggregation of community resilience goals to obtain minimum performance objectives for buildings under tornado hazards. *Struct. Saf.* **2018**, *70*, 82–92. [CrossRef]
5. Kosiba, K.; Wurman, J. The strongest winds in tornadoes are very near the ground. *Commun. Earth Environ.* **2023**, *4*, 1–6. [CrossRef]
6. Hale, S.E.; Gardiner, B.A.; Wellpott, A.; Nicoll, B.C.; Achim, A. Wind loading of trees: Influence of tree size and competition. *Eur. J. For. Res.* **2012**, *131*, 203–217. [CrossRef]
7. Schelhaas, M.; Kramer, K.; Peltola, H.; van der Werf, D.; Wijdeven, S. Introducing tree interactions in wind damage simulation. *Ecol. Model.* **2007**, *207*, 197–209. [CrossRef]
8. Zhu, P. Impact of land-surface roughness on surface winds during hurricane landfall. *Q. J. R. Meteorol. Soc.* **2008**, *134*, 1051–1057. [CrossRef]
9. Lewellen, D.C. Effects of topography on tornado dynamics: A simulation study. In Proceedings of the 26th Conference on Severe Local Storms, Nashville, TN, USA, 5–8 November 2012.
10. Lewellen, D.C.; Lewellen, W.S.; Xia, J. The Influence of a Local Swirl Ratio on Tornado Intensification near the Surface. *J. Atmos. Sci.* **2000**, *57*, 527–544. [CrossRef]
11. Godfrey, C.M.; Peterson, C.J. Estimating Enhanced Fujita Scale Levels Based on Forest Damage Severity. *Weather Forecast* **2017**, *32*, 243–252. [CrossRef]
12. Rhee, D.M.; Lombardo, F.T. Improved near-surface wind speed characterization using damage patterns. *J. Wind. Eng. Ind. Aerodyn.* **2018**, *180*, 288–297. [CrossRef]
13. Peterson, C.J. Consistent influence of tree diameter and species on damage in nine eastern North America tornado blowdowns. *For. Ecol. Manag.* **2007**, *250*, 96–108. [CrossRef]
14. Potvin, C.K.; Broyles, C.; Skinner, P.S.; Brooks, H.E. Improving Estimates of U.S. Tornado Frequency by Accounting for Unreported and Underrated Tornadoes. *J. Appl. Meteorol. Clim.* **2022**, *61*, 909–930. [CrossRef]
15. Kunkel, J.; Hanesiak, J.; Sills, D. The Hunt for Missing Tornadoes: Using Satellite Imagery to Detect and Document Historical Tornado Damage in Canadian Forests. *J. Appl. Meteorol. Clim.* **2023**, *62*, 139–154. [CrossRef]
16. Burow, D.; Herrero, H.V.; Ellis, K.N. Damage Analysis of Three Long-Track Tornadoes Using High-Resolution Satellite Imagery. *Atmosphere* **2020**, *11*, 613. [CrossRef]
17. Budney, L.J. Unique Damage Patterns Caused by a Tornado in Dense Woodlands. *Weatherwise* **1965**, *18*, 74–86. [CrossRef]
18. Fujita, T.T. Tornadoes and Downbursts in the Context of Generalized Planetary Scales. *J. Atmos. Sci.* **1981**, *38*, 1511–1534. [CrossRef]
19. Fujita, T.T. The Teton-Yellowstone Tornado of 21 July 1987. *Mon. Weather. Rev.* **1989**, *117*, 1913–1940. [CrossRef]
20. Lyza, A.W.; Knupp, K.R. An observational analysis of potential terrain influences on tornado behavior. In Proceedings of the 27th Conference on Severe Local Storms, Portland, OR, USA, 3–7 November 2014; p. 11A.1A. Available online: https://ams.confex.com/ams/27SLS/webprogram/Paper255844.html (accessed on 20 December 2023).
21. Karstens, C.D.; Gallus, W.A.; Lee, B.D.; Finley, C.A. Analysis of Tornado-Induced Tree Fall Using Aerial Photography from the Joplin, Missouri, and Tuscaloosa–Birmingham, Alabama, Tornadoes of 2011. *J. Appl. Meteorol. Clim.* **2013**, *52*, 1049–1068. [CrossRef]
22. Cannon, J.B.; Hepinstall-Cymerman, J.; Godfrey, C.M.; Peterson, C.J. Landscape-scale characteristics of forest tornado damage in mountainous terrain. *Landsc. Ecol.* **2016**, *31*, 2097–2114. [CrossRef]
23. Wagner, M.A.; Doe, R.K.; Wang, C.; Rasmussen, E.; Coniglio, M.C.; Elmore, K.L.; Balling, R.C.; Cerveny, R.S. High-resolution observations of microscale influences on a tornado track using Unpiloted Aerial Systems (UAS). *Mon. Weather Rev.* **2021**, *149*, 2819–2834. [CrossRef]
24. Blanchard, D.O. A Comparison of Wind Speed and Forest Damage Associated with Tornadoes in Northern Arizona. *Weather Forecast* **2013**, *28*, 408–417. [CrossRef]
25. Mohammadi, M.E.; Watson, D.P.; Wood, R.L. Deep Learning-Based Damage Detection from Aerial SfM Point Clouds. *Drones* **2019**, *3*, 68. [CrossRef]
26. Liao, Y.; Mohammadi, M.E.; Wood, R.L. Deep Learning Classification of 2D Orthomosaic Images and 3D Point Clouds for Post-Event Structural Damage Assessment. *Drones* **2020**, *4*, 24. [CrossRef]
27. Rhee, D.M.; Lombardo, F.T.; Kadowaki, J. Semi-automated tree-fall pattern identification using image processing technique: Application to alonsa, MB tornado. *J. Wind. Eng. Ind. Aerodyn.* **2021**, *208*, 104399. [CrossRef]
28. Rhee, D.M.; Stevenson, S.; Lombardo, F.T.; Kopp, G. Tornado Wind Speed Estimation Methods in Rural Forested Regions: The Alonsa, MB Tornado. In Proceedings of the 6th AAWE, Clemson, SC, USA, 12–14 May 2021.

29. Chen, Z.; Wagner, M.; Das, J.; Doe, R.K.; Cerveny, R.S. Data-Driven Approaches for Tornado Damage Estimation with Unpiloted Aerial Systems. *Remote Sens.* **2021**, *13*, 1669. [CrossRef]
30. Carani, S.; Pingel, T.J. Detection of Tornado damage in forested regions via convolutional neural networks and uncrewed aerial system photogrammetry. *Nat. Hazards* **2023**, *119*, 143–166. [CrossRef]
31. Reder, S.; Mund, J.-P.; Albert, N.; Waßermann, L.; Miranda, L. Detection of Windthrown Tree Stems on UAV-Orthomosaics Using U-Net Convolutional Networks. *Remote Sens.* **2022**, *14*, 75. [CrossRef]
32. Agisoft. Agisoft Metashape: Agisoft Metashape. Agisoft 2023, 7–9. Available online: https://www.agisoft.com/ (accessed on 22 January 2024).
33. Drone for Fast and Accurate Survey Data Every Time. 2023. Available online: https://wingtra.com/ (accessed on 22 January 2024).
34. Gaffrey, D.; Sloboda, B.; Matsumura, N. Representation of tree stem taper curves and their dynamic, using a linear model and the centroaffine transformation. *J. For. Res.* **1998**, *3*, 67–74. [CrossRef]
35. Darwin. Available online: https://darwin.v7labs.com/datasets/518466/dataset-management (accessed on 22 January 2024).
36. Jocher, G.; Chaurasia, A.; Qiu, J. GitHub—Ultralytics/ultralytics: NEW-YOLOv8 ? in PyTorch > ONNX > OpenVINO > CoreML > TFLite. 2023. Available online: https://github.com/ultralytics/ultralytics (accessed on 22 January 2024).
37. Lou, H.; Duan, X.; Guo, J.; Liu, H.; Gu, J.; Bi, L.; Chen, H. DC-YOLOv8: Small-Size Object Detection Algorithm Based on Camera Sensor. *Electronics* **2023**, *12*, 2023. [CrossRef]
38. Yan, S.; Fu, Y.; Zhang, W.; Yang, W.; Yu, R.; Zhang, F. Multi-Target Instance Segmentation and Tracking Using YOLOV8 and BoT-SORT for Video SAR. In Proceedings of the 2023 5th IEEE International Conference on Electronic Engineering and Informatics (EEI 2023), Wuhan, China, 30 June–2 July 2023; pp. 506–510. [CrossRef]
39. Kong, G.; Dong, L.; Dong, W.; Zheng, L.; Tian, Q. Coarse2Fine: Two-Layer Fusion For Image Retrieval. 2016. Available online: http://www.michaelshell.org/contact.html (accessed on 12 January 2024).
40. Workspace Home. Available online: https://app.roboflow.com/richard-wood-university-of-nebraska-lincoln (accessed on 23 January 2024).
41. Shorten, C.; Khoshgoftaar, T.M. A survey on Image Data Augmentation for Deep Learning. *J. Big Data* **2019**, *6*, 60. [CrossRef]
42. Poojary, R.; Raina, R.; Mondal, A.K. Effect of data-augmentation on fine-tuned CNN model performance. *IAES Int. J. Artif. Intell. (IJ-AI)* **2021**, *10*, 84–92. [CrossRef]
43. Lopes, R.G.; Yin, D.; Poole, B.; Gilmer, J.; Cubuk, E.D. Improving Robustness without Sacrificing Accuracy with Patch Gaussian Augmentation. *arXiv* **2019**. Available online: https://arxiv.org/abs/1906.02611v1 (accessed on 20 March 2024).
44. Gómez-Reyes, J.K.; Benítez-Rangel, J.P.; Morales-Hernández, L.A.; Resendiz-Ochoa, E.; Camarillo-Gomez, K.A. Image Mosaicing Applied on UAVs Survey. *Appl. Sci.* **2022**, *12*, 2729. [CrossRef]
45. Yue, Y.; Finley, T.; Radlinski, F.; Joachims, T. A support vector method for optimizing average precision. In Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Taipei, Taiwan, 23–27 July 2023; Cornell University: Ithaca, NY, USA, 2007; pp. 271–278. [CrossRef]
46. Ali, L.; Alnajjar, F.; Parambil, M.M.A.; Younes, M.I.; Abdelhalim, Z.I.; Aljassmi, H. Development of YOLOv5-Based Real-Time Smart Monitoring System for Increasing Lab Safety Awareness in Educational Institutions. *Sensors* **2022**, *22*, 8820. [CrossRef] [PubMed]
47. Soares, M.M.; Rosenzweig, E.; Marcus, A. International Conference on Human-Computer Interaction. Design, User Experience, and Usability: UX Research, Design, and Assessment. In Proceedings of the 11th International Conference, DUXU 2022 held as part of the 24th HCI International Conference, HCII 2022, Virtual Event, 26 June–1 July 2022.
48. Larsen, D.R. Simple taper: Taper equations for the field forester. In Proceedings of the 20th Central Hardwood Forest Conference, Columbia, Columbia, MO, USA, 28 March–1 April 2016; US Forest Service: Washington, DC, USA, 2016; pp. 265–278.
49. Sills, D.; Kopp, G. *Northern Tornadoes Project. Annual Report 2021 v2.*; Western Libraries: Yarmouth, NS, USA, 2022. [CrossRef]
50. National Weather Service. The Violent Tornado Outbreak of December 10–11. 2021. Available online: https://www.weather.gov/pah/December-10th-11th-2021-Tornado (accessed on 26 January 2024).
51. Data Downloads | State of Tennessee Elevation LiDAR Project. Available online: https://lidar.tn.gov/pages/data-downloads (accessed on 3 March 2024).
52. Satrio, M.A.; Bodine, D.J.; Reinhart, A.E.; Maruyama, T.; Lombardo, F.T. Understanding How Complex Terrain Impacts Tornado Dynamics Using a Suite of High-Resolution Numerical Simulations. *J. Atmospheric Sci.* **2020**, *77*, 3277–3300. [CrossRef]
53. Sterling, M.; Huo, S.; Baker, C. Using crop fall patterns to provide an insight into thunderstorm downbursts. *J. Wind. Eng. Ind. Aerodyn.* **2023**, *238*, 105431. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.