remote sensing

MDPI

Article

Fast Tree Detection and Counting on UAVs for Sequential Aerial Images with Generating Orthophoto Mosaicing

Pengcheng Han [1,2], Cunbao Ma [1], Jian Chen [3], Lin Chen [1], Shuhui Bu [1], Shibiao Xu [4,*], Yong Zhao [1], Chenhua Zhang [5] and Tatsuya Hagino [5]

[1] School of Civilaviation, School of Aeronautics, Northwestern Polytechnical University, Xi'an 710072, China
[2] Xi'an Vision Huitu Intelligent Technology Co., Ltd., Xi'an 710000, China
[3] Energy and Power Engineering School, Xi'an Jiaotong University, Xi'an 710000, China
[4] School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing 100876, China
[5] Graduate School of Media and Governance, Keio University, Tokyo 108-8345, Japan
* Corresponding: shibiaoxu@bupt.edu.cn

Abstract: Individual tree counting (ITC) is a popular topic in the remote sensing application field. The number and planting density of trees are significant for estimating the yield and for futher planing, etc. Although existing studies have already achieved great performance on tree detection with satellite imagery, the quality is often negatively affected by clouds and heavy fog, which limits the application of high-frequency inventory. Nowadays, with ultra high spatial resolution and convenient usage, Unmanned Aerial Vehicles (UAVs) have become promising tools for obtaining statistics from plantations. However, for large scale areas, a UAV cannot capture the whole region of interest in one photo session. In this paper, a real-time orthophoto mosaicing-based tree counting framework is proposed to detect trees using sequential aerial images, which is very effective for fast detection of large areas. Firstly, to guarantee the speed and accuracy, a multi-planar assumption constrained graph optimization algorithm is proposed to estimate the camera pose and generate orthophoto mosaicing simultaneously. Secondly, to avoid time-consuming box or mask annotations, a point supervised method is designed for tree counting task, which greatly speeds up the entire workflow. We demonstrate the effectiveness of our method by performing extensive experiments on oil-palm and acacia trees. To avoid the delay between data acquisition and processing, the proposed framework algorithm is embedded into the UAV for completing tree counting tasks, which also reduces the quantity of data transmission from the UAV system to the ground station. We evaluate the proposed pipeline using sequential UAV images captured in Indonesia. The proposed pipeline achieves an F1-score of 98.2% for acacia tree detection and 96.3% for oil-palm tree detection with online orthophoto mosaicing generation.

Keywords: aerial images; tree counting; mosaicing orthophoto; point supervised; UAV

Citation: Han, P.; Ma, C.; Chen, J.; Chen, L.; Bu, S.; Xu, S.; Zhao, Y.; Zhang, C.; Hagino, T. Fast Tree Detection and Counting on UAVs for Sequential Aerial Images with Generating Orthophoto Mosaicing. *Remote Sens.* **2022**, *14*, 4113. https://doi.org/10.3390/rs14164113

Academic Editors: Weijia Li, Lichao Mou, Angelica I. Aviles-Rivero, Runmin Dong and Juepeng Zheng

Received: 24 July 2022
Accepted: 18 August 2022
Published: 22 August 2022

1. Introduction

Automatic tree detection and counting using high-resolution remote sensing images has attracted the attention of a large number of researchers in recent decades. Information about the shape, localization, and health of trees in plantation areas are essential for intelligent agriculture to monitor the quality of tree growth. The image acquisition usually comes from satellites and Unmanned Aerial Vehicles (UAVs). Due to the rapid development of satellite and UAV technologies, abundant data have further promoted the development of related tree detection and counting technologies.

In general, most existing methods of tree detection and counting are concentrated on satellite images [1–5]. Firstly, the whole large scale remote sensing image is split into hundreds of patches using the sliding window technique. Then classification or detection-based methods are used to detect and localize the targets. Finally, the detection results

from individual patches are fused into the whole remote sensing image. The satellite technique is a powerful tool for monitoring tasks, including land cover change detection, fire monitoring, urban construction, natural disaster monitoring, etc. Recently, UAVs have become promising tools to obtain high quality images with low-cost, lightweight and high frequency usage, which have potentials for various analysis and monitoring domains, including environmental monitoring and analysis, precise and intelligent agriculture, and disease detection [6]. However, what UAVs acquire with camera devices is a series of image data, and sequence image detection could result in a lot of duplication, which is not convenient and reduces detection efficiency and quality in large scale regions. Therefore, aerial image mapping is a necessary task before image analysis.

In recent decades, aerial image stitching has been used in lots of situations, such as agricultural plant protection, forest fire monitoring, forest tree counting, post-disaster assessment and military investigations. Generally, there are two ways to realize aerial image mosaicing tasks. One is offline mosaicing [7–9], where whole images of a target region captured with unmanned aerial vehicles (UAVs) are processed and optimized together. This approach requires integrated necessary information of captured images for image mapping, which generally leads to accurate stitching results. The other is online mosaicing, with real-time estimates [10–12]. It is necessary in some special application scenarios like post-disaster rescue, military reconnaissance and tree asset statistics, which are situations that place high demand on time efficiency. In general, the major differences between online and offline mosaicing are camera pose estimation algorithms and the 3D point cloud generation method. Structure from motion (SfM) [13,14] and simultaneous localization and mapping (SLAM) [15–18] are two classic methods used to estimate the camera pose and generate a point cloud. Simultaneous localization and mapping (SLAM) is the technique to construct or update a map of an unknown environment while simultaneously keeping track of an agent's location within it. Structure from motion (SfM) is a photogrammetric range imaging technique for estimating three-dimensional structures from two-dimensional image sequences that may be coupled with local motion signals. After the camera pose and point cloud are obtained with one of above-mentioned techniques, the captured images are then projected to the correct position with homography transformation methods. Finally, these discrete pictures are fused to form a whole mosaic. However, SfM methods [13,14,19,20] are basic offline algorithms with expensive computation, which places high demand on hardware requirements. Therefore, it is inconvenient to be deployed on edge devices. By the way, long computations will reduce the overall work efficiency, potentially increases the time cost, especially for large-scale planting statistics introduced in this paper. For most SLAM methods [10], a bundle adjustment algorithm is used to estimate camera poses and refine landmark positions; landmark positions are the key feature points jointly observed with multiple images. Although the SLAM-based real-time system is faster than the SfM-based mapping system, the step of bundle adjustment is still time-consuming for real-time construction systems and the matching performance is poor under low overlap conditions. To solve these problems, a multiplanar hypothesis-based pose optimization method is proposed to estimate camera poses and generate mosaicing simultaneously. This could accelerate the calculation speed and achieve robust stitching performance with sequential low overlap images in the embedded devices.

For tree counting and detection tasks, most of the early works [21,22] focused on feature representation design. Some good handcrafted features like Scale invariant feature transform (SIFT) and histogram of oriented gradient (HOG) were firstly used to extract key points. Then classifiers like support vector machine (SVM) [22,23], Random Forests [24], K-Means [25] and extreme learning machine (ELM) were conducted to accomplish target trees detection task using above-mentioned traditionally handcrafted features. Nevertheless, the detection scenes are relatively simple; the trees are sparsely distributed and clearly outlined with no canopy overlapping. Thus, the traditional handcrafted features-based methods could conduct the task with good performance. However, these methods are not end-to-end frameworks, and they consist of several separate algorithm steps that could

bring extra unexpected noise or errors because of careless data transport between separate algorithms. In addition, handcrafted feature engineering requires expensive computation and efforts that depend on experts knowledge which is not easily obtained for various intelligent tasks.

Recently, because of the powerful capability of learning features, convolutional neural networks (CNNs) [26] have achieved great success in various fields including computer vision, natural language processing, medical diagnosis, and so forth [5,27–32]. Existing works could be divided into detection-based, segmentation-based, and regression-based methods. For detection-based supervised methods, researchers collect large amounts of tree data for deep learning (DL) algorithms to learn meaningful features for tree counting or other object detection task [1,5,33–38]. In [2,6,39–41], CNN framework and full supervised Faster R-CNN [42] are designed for oil-palm trees or other fruit detection with bounding box annotations. For segmentation-based supervised methods, some U-Net [43] and Fully Convolutional Neural Networks (FCNs) [32]-based detection works [44,45] are used to estimate the presence map of trees with remote sensing images, which generate the final results with mask that contains contour and size of tree. However, the "box in, box out" detection-based or "mask in, mask out" segmentation-based methods require lots of careful annotations that are time-consuming and laborious because of irregular target shape and complicated detection scenes. In order to alleviate the heavy annotation problem, the "point in, point out" regression-based methods have attracted attention. In [46,47], the researchers simultaneously predict the localization and number of persons with estimating the density maps using point annotation information. However, these methods only predict the center point while ignoring the size and contour of target objects. To fast count the trees with high quality on UAVs, a weakly supervised deep learning-based tree counting framework is proposed in this paper, which could avoid expensive bounding box or mask annotation costs. Not only the accurate number of trees but also the mask of objects could be obtained; the results are comparable to those of strongly supervised methods [42].

In summary, we propose an aerial orthophoto mosaicing-based tree counting framework to realize fast completed automatic tree detection and counting in this paper. The tree detection and counting task is completed with a planar restricted pose graph optimization-based integrated real-time orthoimage generation and point-based supervised training framework. Therefore, the proposed pipeline could be able to improve the accuracy and speed of mosaicing and tree counting, and it is efficient and robust for board devices. The main contributions are as follows:

1. **A novel efficient tree detection and counting framework for UAVs** : Compared to the current tree counting pipeline, our method provides a real-time solution for detection tasks with UAVs. High-quality mosaicing is efficiently generated with less calculations; detection and counting task is completed with fast annotation, training and inference analysis pipelines.
2. **A multiplanar hypothesis-based online pose optimization**: A multiplanar hypothesis-based pose optimization method is proposed to estimate camera poses and generate mosaicing simultaneously. The number of parameters about reprojection error is effectively reduced; the method could accelerate the calculation speed and achieve robust stitching performance with sequential low overlap images in the embedded devices.
3. **Point-supervised-based attention detection framework**: A point supervised method could not only estimate the localization of trees but also generate a contour mask that is comparable to full supervised methods. The supervised label information is easy to be obtained, which could be effective for entire learning framework.
4. **An embedded system with a proposed algorithm on UAVs**: An embedded fully automatic system is embedded into the UAV for completing integrated stitching and tree counting tasks; the whole procedure requires no human intervention at all. In addition, buildings or trees could have a greater negative impact on the communication link between the UAV and a ground station; the embedded system could ignore this negative effects and improve work efficiency.

The remainder of this article is organized as follows. Section 2 briefly reviews the related work. In Section 3, the two-stage vision framework is illustrated in detail. Experimental results and discussions are presented in Section 4. Finally, we draw conclusions about the proposed fast tree counting pipeline in Section 5.

2. Related Work

2.1. Image Mosaicing

In general, a global positioning method like SfM is the best choice for image mosaicing, which is designed to reconstruct a digital model from abundant multi-view frames to guarantee the stitching quality. In the past decades, many kinds of SfM methods like incremental [13,14], hierarchical [48] and global [19,20] methods have been proposed to reconstruct various scenes in detail. The general process of stitching with the above-mentioned methods is pose optimization, 3D point cloud generation, ground control point(GCP) identification and finally moscaicing all frames [49]. The procedure is time consuming and costs several hours in real applications. In order to accelerate the procedure of SfM-based methods, an efficient offline aerial frame stitching approach [9] with a planar constrained strategy is adopted to optimize pose, which guarantees robust performance and fast speed.

However, it is still not suitable for real-time and incremental application fields. Thus, SLAM-based methods [11] are designed for image mosaicing with incremental stitching. Map2Dfusion [10] obtains real-time position by synchronizing the global navigation satellite system (GNSS) information and video streams, which overcomes global drift, and then generates orthomosaic with weighted pyramid fusion. Lati et al. [50] introduced a fuzzy clustering-based algorithm to remove outliers to improve optimization speed. In [51], sparse bundle adjustment (BA) is designed to speed up the pose optimization in the SLAM reconstruction procedure, but it does not easily guarantee reconstruction quality. Therefore, these traditional SLAM-based algorithms could efficiently estimate poses and generate 3D point cloud simultaneously, but they are still difficult to be deployed in embedded devices to complete stitching tasks with processing low overlapping sequential images.

Considering that the flying height of UAVs is generally in the order of hundreds of meters or even thousands of meters, it is recommended to assume the scene as multi-planar in the tree counting task. Based on such assumptions, the algorithm could be faster to compute and consume less computational effort.

2.2. Tree Counting

To achieve a high-quality tree counting task, over the past few decades, researchers have explored lots of efficiency algorithms, which are mainly summarized into the following two categories: the traditional image analysis-based algorithms and deep learning-based methods. The first category includes image binarization, template matching, the local maximum filter-based methods, maximum likelihood, support vector machine, extreme learning machine, random forest, etc. In these traditional methods [21–25], feature engineering is essential which is a complex, time-consuming process that needs to be redesigned whenever the problem or the dataset changes. Thus, feature engineering relies on expert knowledge with lots of effort and the generalization is not enough for various tasks.

The second category is the powerful deep learning-based algorithm, which includes convolutional neural network (CNN) classification, object detection, semantic segmentation and regression methods. Some state-of-the-art CNN classification and object detection models are as follows. Li et al. [2] train the convolutional neural network (CNN) to predict the tree labels with collecting large amounts of annotated data. They split the whole big image into hundreds of patches through the sliding window technique to detect the target tree with AlexNet, VGG, and ResNet CNN architectures. They propose a two-stage convolutional neural network [3] further improving crown detection accuracy. With the development of deep learning detection architecture, lots of end-to-end object detection-based methods like YOLO architecture [52], Faster R-CNN [53], and Mask R-

CNN [54] have been designed for the tree counting task. Zheng et al. [6] propose a faster R-CNN-based method called as Multi-class Oil-Palm Detection approach (MOPAD) to accurately detect and monitor localization and growing status of trees. In order to better apply the deep learning technology, Li et al. [4] propose the first Field Programmable Gate Array (FPGA)-based real-time tree detection method for large-scale remote sensing images. In addition, some segmentation models like U-Net [55,56], DeepLab v3+ [57]and FCN [58,59], and other advanced models have been applied in tree detection task and achieve promising results. Unlike object detection-based methods, these methods require more time-consuming annotations, what's more, some inaccurate labelling could bring wrong predictions because of irregular target shape and complicated prediction background information.

From the above-mentioned deep learning frameworks [1,2,4–6,33,34,36–41,44,45], one of the most significant advantages of using deep learning techniques is that a deep learning-based method is an end to end framework, which does not require feature engineering and learn discriminative features itself through training. However, existing tedious annotation is inevitable; "point in, box or mask out" algorithms are proposed to speed up the deep learning-based tree counting applications.

3. Methodology

3.1. Overview

The whole proposed system is an integrated vision framework named real-time orthophoto mosaicing-based tree counting pipeline illustrated in Figure 1, which includes "real-time stitching" and "fast tree counting" modules. In the stitching step, sequential images with GNSS information are obtained through the UAV Software Development Kit (SDK) and posted to the algorithm during flight mission. After the interesting region of map detection, a fast detection algorithm is used to complete the tree counting task. There are mainly two advantages: (1) we only use a series of image data captured to detect trees, which will result in a lot of duplications because of overlap among images; this is not convenient and reduces detection efficiency and quality in large scale regions. A real-time stitching module could quickly reconstruct the region areas with high quality for the next tree counting analysis, which could improve the detection performance with a global perspective. (2) After the orthophoto mosaicing is generated, it is cut into tiles, this data format is suitable for the next tree detection with parallel computation. Thus, the calculation efficiency of the entire workflow is high, and it could be easily connected to a geographic information system (GIS) for visualization or deployed into embedded devices like UAVs.

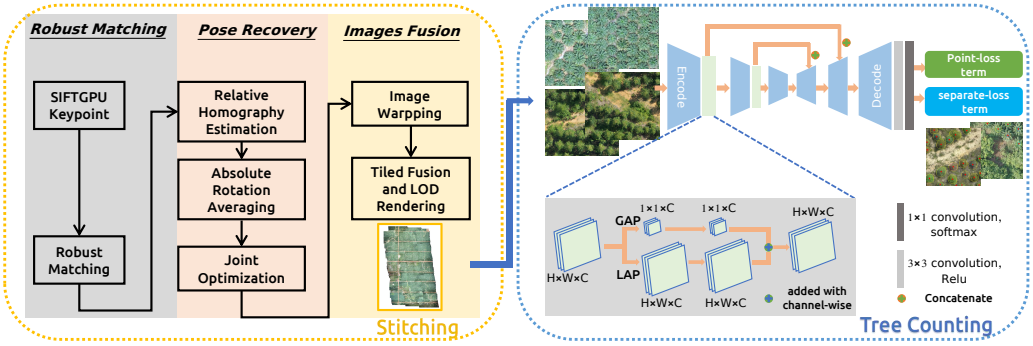


Figure 1. The algorithm presented in this paper is an integrated vision framework, which consists of "real-time stitching" and "fast tree counting" modules. This framework could be easily applied into the embedded system for UAVs; it makes the UAV support online georeferenced large-scale low-overlap aerial images stitching and fast tree counting tasks, which could greatly improve work efficiency.

In the stitching procedure, when the algorithm receives real-time captured images, it extracts the key-points and inserts them into a matching graph using appearance and spatial-based faster neighbor query and matching algorithms. After the matching operation,

the algorithm computes absolute rotation and translation by aligning the visual position with the GNSS position. While in the optimization step, the error in the graph propagates through homograph matrix. The propagation, which is-based on homograph matrix is robust and efficiency, making it possible to handle tough sequences whose overlap is less than 50%. Every frame with pose is delivered to stitching thread. In addition, our orthomosaic is fused with weighted multi-band laplacian pyramid, which makes it seamless and smoother. Considering the large-scale reconstruction and smooth rendering, the algorithm balances memory and hard disk useage, using an Least Recently Used (LRU) to manage multiple zoom level tiles, which ensures the invariance of the shape of the object, as well as the correctness of the direction and mutual position, so it is often used in navigation and aviation. A Level of Detail (LoD) algorithm is adopted to update the multiple zoom level tiles, which could increase the efficiency of rendering with decreasing the workload on graphics computation procedure.

In the detection procedure, in order to effectively predict the localization of trees for large scale scenes, and avoid expensive bounding box or mask annotations, a weakly supervised attention method only with point annotations that could reduce human annotation labor is introduced in this section. The proposed weakly semantic segmentation network not only predicts the positions of trees but also obtains the outline of trees. The U-Net framework is an encoder–decoder architecture consisting of contracting path and expanding path designed to capture context and extract discriminative feature representations. Inspired by this, a novel point-supervised attention network is proposed, which is pre-trained with RestNet-50 with first four encoder layers to obtain the feature representation, which could be further improved with spatial unit (SU) consisting of global average pooling (GAP) and local average pooling (LAP) operations for the tree detection task. In the training step, point supervised loss function consisting of point-loss and separate-loss terms, which are used to guide the network to generate the mask of trees with point supervised information. After the network is trained completely, the weights of the proposed network are fixed and deployed in edge device like UAV for tree detection. For the convenience of deployment and application, we train two sets of weights for acacia and oil-palm detection, respectively. When we apply the pipeline for new tree species, several hundreds of annotated data are needed to train the network and generate the new weights. In the inference step, firstly, the orthophoto mosaicing is real-time-generated with sequential images captured in UAV; secondly, the generated orthophoto mosaicing is cut into thousands of tiles and served in the form of Hyper Text Transfer Protocol (HTTP). A well-trained network is used to obtain the localization and number of instances for tree detection task with each tile from interesting regions.

3.2. Real Time Generating Orthophoto Mosaicing

The traditional SLAM methods proposed optimize poses that are-based on GNSS and multiplanar hypothesis, which ensures robustness under low overlap. Like SLAM system-based on feature points, we choose SIFT for its stability. For efficiency, the OpenGL Shading Language (GLSL)-based SIFT algorithm is used to extract the feature points of the image, and GLSL requires a GPU that has large memory and supports dynamic branching to accelerate the computation of SIFT description. The RootSIFT algorithm [60] is adopted to generate the descriptors to further improve the matching quality and enhance the robustness. After calculating the pose, we project the photo onto a two-dimensional map according to the optimized plane, and adopt LoD for smooth rendering.

3.2.1. Keypoint Detection and Matching

Considering that SIFT algorithm contains a large number of independent pixel wise processes, the GLSL accelerated SIFT is performed to improve the computing speed and platform independency. That is to say, the algorithm has the potential to be deployed in embedded or mobile terminals. The matching process adopts the RootSIFT method to normalize the descriptor and calculate the square root to measure the distance between two

SIFT feature points. Some studies have proven that this method is better than the method of directly using Euclidean distance to measure similarity.

3.2.2. Online Planar Restricted Pose Recovery

In Figure 2, unlike the general full-scene plane assumption, our proposed method adopts the multiplanar assumption, that is, each image captured is assumed to be flat. The general full-scene plane assumption could result in larger estimation bias. This multiplanar assumption processing method weakens the shortcoming of full-scene plane assumption and improves the generalizability of the method for aerial image stitching.

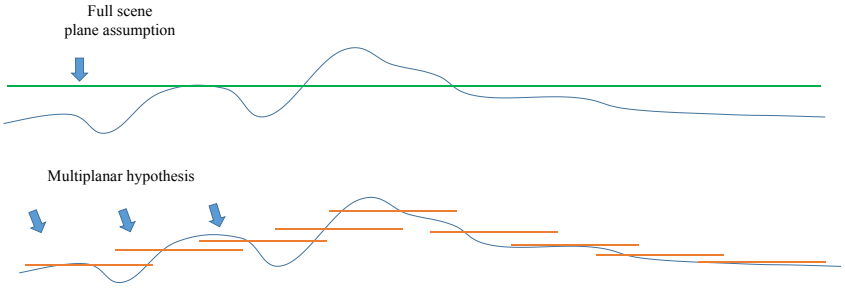


Figure 2. Blue curve represents the real terrain, green line represents the full-scene plane assumption, and orange lines represent multiplanar assumption.

When the overlap is low, the traditional PnP-based pose tracking cannot work properly, due to the lack of common view; tracking is equivalent to the initialization. To accurately estimate camera poses, we first decouple and estimate the rotation and translation separately, adopting rotation and translation averaging algorithm [61]. Then the GNSS information is fused in the rotation averaging procedure to make the system more stable. Unlike the general bundle adjustment algorithm, the graph optimization procedure depends on the homograph propagation and does not need triangulation, and reprojection errors are not constructed with map points, which accelerate the mapping speed and improve the robustness of system. For image i, the camera poses and height will be composed of a seven-DOF parameter, $\mathbf{T}_i = (\mathbf{R}_i, \mathbf{t}_i, h_i)$, $\mathbf{R}_i$, $\mathbf{t}_i$ and h_i are the rotation, translation matrix and flight height for image i, respectively. The submap $\mathbf{S}$ is obtained according to the matches and geodetic position. The optimization function is as follows:

$$\hat{\mathbf{T}} = \arg\min \sum_{i,j \in \mathbf{S}} \sum_{m} \xi_m(\mathbf{T}_i, \mathbf{T}_j) + \sum_{i \in \mathbf{S}} \tau(\mathbf{T}_i, \mathbf{g}_i) \tag{1}$$

where $\tau(\mathbf{T}_i, \mathbf{g}_i) = \mathbf{t}_i - \mathbf{g}_i$ is the GNSS prior factor, which is an error loss used to optimize the camera pose with considering the GNSS information. $\mathbf{g}_i$ is the local cartesian coordinates computed from the GNSS information for each frame, while $\xi_m(\mathbf{T}_i, \mathbf{T}_j)$ is the reprojection error of match $m(\mathbf{u}_i, \mathbf{u}_j)$ between frames i and j, it is defined as:

$$\mathbf{a} = \mathbf{R}_i \cdot Proj^{-1}(\mathbf{u}_i) \tag{2}$$

$$\mathbf{p} = \mathbf{t}_i + \frac{(h_i + h_j)/2 - \mathbf{t}_i^z}{\alpha^z} \mathbf{a} \tag{3}$$

$$\xi_m(\mathbf{T}_i, \mathbf{T}_j) = Proj(\mathbf{R}_j^{-1} \cdot (\mathbf{p} - \mathbf{t}_j)) - \mathbf{u}_j \tag{4}$$

Here, $\mathbf{a}$ is the unit vector represents the view direction of keypoint $\mathbf{u}_i$ and p is the 3D coordinates of the keypoint. h_i, h_j are the local ground height estimation of image i, j.

Proj is denoted as the pinhole projection in camera coordinates, and $Proj^{-1}$ is the inverse projection in the plane where $z = 1$:

$$\mathbf{u} = Proj(\mathbf{p}) = \left(\frac{x * f_x}{z} + c_x, \frac{y * f_y}{z} + c_y\right)^T \quad (5)$$

$$\hat{\mathbf{p}} = Proj^{-1}(\mathbf{u}) = \left(\frac{x - c_x}{f_x}, \frac{y - c_y}{f_y}, 1\right)^T \quad (6)$$

The Ceres Solver library (http://ceres-solver.org, accessed on 1 March 2022) is a large scale non-linear optimization library that is used to deal with the graph optimization problem in the proposed pipeline.

3.2.3. Georeferenced Images Fusion with Tiling and LoD

After the camera pose estimation, the original images are reprojected onto the digital orthophoto map (DOM). Due to the overlapping of captured sequential images, the adaptive weighted multiband algorithm [10] is adopted to fuse the images. The generated mosaicing consists of several level pyramid tiles with LoD, which could increase the efficiency of rendering with decreasing the graphics workload for visualization. In addition, a thread-safe LRU technique is used to cache the segmented image tiles to decrease memory cost and a thread pool is used to process the tiles in parallel for further speed acceleration.

3.3. Weakly Supervised Attention Counting Tree Network

After orthophoto generation, we cut the oil-palm and acacia orthophoto into thousands of tiles, and split the collected tile images into training and testing sets. In the following experiment part, the training dataset consists of 2880 images of acacia and 2280 images of oil-palm, and the test dataset consists of 1920 images of acacia and 1520 images of oil-palm that are used to test the proposed method's effectiveness. The training set is labeled with labelme toolbox [62] by point annotation. Then the labelled data is fed into a weakly supervised attention counting tree network to learn the feature representation of target tree. The network consists of attention-based tree feature extractor and point supervised loss function described as follows.

3.3.1. Attention Based Tree Feature Extractor Network

In the attention-based tree feature extractor network (ATFENet), the U-net similar CNN architecture, ResNet-50 is adopted as the network backbone to complete the feature extractor procedure. The contracting path and expanding path are combined through skip-connections [43] to capture abundant context and learn to extract discriminative feature representations for localization task. The size of input original images and output prediction mask are both N×W×H. Thus, The output array is a pixel-by-pixel semantic mask predicted with 1×1 convolution and softmax function in the last layer of network to determine which pixel is target tree or not.

Inspired by the human visual perception system, attention mechanism constructs the dependent relationships between the input and target information, has been tried in many intelligence-related tasks, like scene parsing, sentences translation and so forth [63]. An attention mechanism could quickly extract important features from the sparse data, while a self-attention mechanism is an improvement of attention mechanism, which reduces the dependence on external information and could better capture the internal correlation of data or features. Therefore, we design an attention-based tree feature extractor network, which captures the rich global and local relationships to effectively model tree detection-related feature relationships. Concretely, we introduce a spatial unit (SU) that consists of global average pooling (GAP) and local average pooling (LAP) operations, which could exploit the global and local spatial channel-wise interdependencies to simultaneously obtain large-size and small-size properties. The global average pooling (GAP) operation is performed on the encode's output feature map to exploit the information with a global

perspective in the first branch. The local average pooling (LAP) with a kernel size $n \times n$ of and a stride of 1 is performed to exploit the informativeness with a local perspective, which computes the average values of sub region whose size is $n \times n$ and traverses the feature map with a sliding window step size of 1. A point-wise convolution with a kernel size of 1×1 is adopted to exploit the channel-wise interdependencies. Through the above operations, the spatial features are fully exploited to quality-boost the basic feature maps for the tree detection task.

3.3.2. Point Supervised Loss Function

The point supervised loss function consists of two terms, which are point level loss and separate loss. Point level loss is a semantic segmentation method designed by Bearman et al. [64], and is used to predict each pixel class according to the point annotations. The separate loss could predict the tree mask and split the masks into individual tree according to the point annotations. The proposed point supervised loss function could be written as below:

$$\mathcal{L}_{all} = \mathcal{L}_p + \mathcal{L}_s \tag{7}$$

Point level loss term. Because we provide point supervised information for training, labels of the pixels corresponding to the supervised points are tree classes, and the other pixel labels are 0. The point annotation labels are used to optimize the cross-entropy function of the model. $\mathcal{I}_f$ denotes the pixels of the image and G_i is the class label of pixel i. The s_{ic} is the score of pixel i belonging to class c, and the softmax function at pixel i is $S_{ic} = exp(s_{ic}) / \sum_{k=1}^{N} exp(s_{ik})$. The point level loss is defined as follows:

$$\mathcal{L}_p(P, S) = -\sum_{i \epsilon \mathcal{I}_f} P_i * log(S_{iG_i}) - \sum_{i \epsilon \mathcal{I}_f} (1 - P_i) * log(1 - S_{iG_i}) \tag{8}$$

where, the S is the output of softmax function, P denotes the tree ground truth label, where N ($N \geq 0$) is the class number of target trees, P_i is 1, 2, ..., n ($1 \leq n \leq N$) correspondingly.

Separate loss term. The point level loss could determine which pixel belongs to a tree class or background. However, the inference model could encounter a condition that two or more trees in one mask area, which decrease the performance of tree detection and counting tasks. Therefore, a separate loss term is designed to divide the mask containing many target objects into individual trees according to the annotations. The watershed algorithm [65] is used to generate individual object blob from the whole tree blobs. The L is denoted as the set of pixels representing the boundaries generated with watershed method.

$$\mathcal{L}_s(S, L) = -\sum_{i \epsilon L} \lambda_i log(S_{i0}) - \sum_{i \epsilon M_{fp}} log(S_{i0}) \tag{9}$$

This term guarantees the tree masks only contain single target tree. In the formula, S_{i0} is the background probability at pixel i, λ_i is the number point annotations ground truth in each image, M_{fp} is the pixels regarded as the tree with training model but there is not any ground-truth point annotation in the pixels. The predictions located in M_{fp} are considered as false positives or unlabelled predictions. For example, the wrong predicted masks shown in Figure 10c are typical false positives, and the second part of this loss could guide this loss term results reduce less false positive tree predictions so as to improve the detection precision.

3.4. Application with Fast Orthophoto Mosaicing and Tree Counting

We design the interface and software development kit (SDK) to embed the integrated tree counting pipeline into fixed wing drones. Nvidia Jetson Xavier NX could ad super computation performance in small devices. The powerful accelerated computing ability is enough to run deep learning-based neural networks in parallel for various AI application systems. Thus, we insert the integrated vision framework into the Jetson Xavier NX device with using a drone interface obtaining camera images and drone status parameters. When

the region of plantation needs to be investigated, a way-point mission is automatically generated with flight height and the overlap rate of captured sequential images. The aerial orthophoto displayed in real time is incrementally updated with processing sequential data.Then a well trained tree counting neural network is used to detect interesting trees with generated orthophoto mosaicing.

This fully automatic integrated perception and analysis system requires no human intervention at all. There are mainly two advantages for UAV-related applications: on one hand, image mosaicing quality depends on the captured images quality, the real-time-embedded system on UAVs could guarantee the quality of data captured with a camera and ignore the negative impacts of communication links, which is suitable for drone operations in large-scale and complex areas. On the other hand, fully autonomous operation greatly improves work efficiency, which greatly reduces time, money and labor costs. Thus, the proposed embedded integrated stitching and tree counting pipeline improves the intelligence of the entire process and greatly reduces costs on the premise of ensuring quality.

4. Experiments

In this experimental section, we focus on the integrated two-stage vision framework with qualitative and quantitative analyses. First, we compare our real time generated mosaicing to the state-of-the-art software Pix4DMapper for evaluating the effectiveness of stitching aerial images algorithm. Second, we create the oil-palm and acacia datasets obtained from fixed-wing UAVs and compare the proposed weakly supervised tree counting algorithm with other state-of-the-art methods. The experimental hardware configuration is as follows: Intel i7-9750H central processing unit (CPU), 32G random-access memory (RAM), and Nvidia GeForce RTX 2060.

4.1. Results of Generating Orthophoto Mosaicing and DOM Quality Comparison

4.1.1. Comparision Experiments

To evaluate the applicability of the proposed method to tree counting task, we collect hundreds of oil-palm sequential images from fixed-wing UAVs. Our algorithm could process the acacia and oil-palm sequential data to generate mosaicing in real-time. The overlapping rate between images is key for the quality of the stitching task. In order to guarantee the overlapping rate, the criteria to consider for a new frame to be added to the mosaic is related to the parameters of flight speed is V (m/s), flight height is h (m), horizontal field of view (FOV) is θ, the overlapping rate is α, the interval of taking pictures is t (s), from the Figure 3, the above-mentioned parameters meet the formula 10.

$$V < \frac{(1-\alpha) * 2h * tan(\theta/2)}{t} \tag{10}$$

For collected oil-palm dataset, the flight height is 430 m, UAV speed is 26 m/s, ground sampling distance (GSD) of the images is 8 cm, the overlapping rate is 90%, the focal length is 5297 pixels, principal point x is 2928 pixels, the principal point y is 1683 pixels. For collected acacia dataset, the flight height is 255 m, UAV speed is 24 m/s, ground sampling distance (GSD) of the images is 3cm, the overlapping rate is 80%, the focal length is 8359 pixels, principal point x is 2979 pixels, the principal point y is 1979 pixels.

The typical overview results of acacia and oil-palm datasets are shown in Figures 4 and 5 with some detailed screenshots. From the figures, we could find that the stitching procedure could be able to generate live mosaicing with high quality, which is comparable to, and even better than, that of the offline processing commercial software Pix4DMapper.

The multi-planar assumption is used in the constrained graph optimization step for the stitching pipeline, and all images are fused and stitched with an adaptive weighted multiband rendering algorithm and homography projection with no triangulation operation. Therefore, the lines in the sequential images could be kept straight easily with the proposed method. By contrast, the orthophoto map obtained from Pix4DMapper are

usually constructed with a 3D triangulation operation, which easily result in deformation or blur of results, as demonstrated in Figure 5. The blue bounding box of Pix4DMapper shows the blur condition. By the way, the stitching procedure of Pix4DMapper consists of aerial triangulation, point cloud densification, mesh generation, mapping and orthophoto generation, this offline method could often achieve great results in the case of abundant image data. However, when the overlap rate of the images is low, there are two problems that cannot be ignored. On the one hand, the lack of GNSS position information makes it difficult for the algorithm to jointly locate the adjacent orbits for the positioning process. On the other hand, smaller side overlapping rate leads to more errors in densification. The combination of the two aspects will result in poor mosaicing. our proposed algorithm combines GNSS information to optimize the pose, and removes redundant degrees of freedom-based on the plane assumption, resulting in high positioning robustness in the stitching procedure.

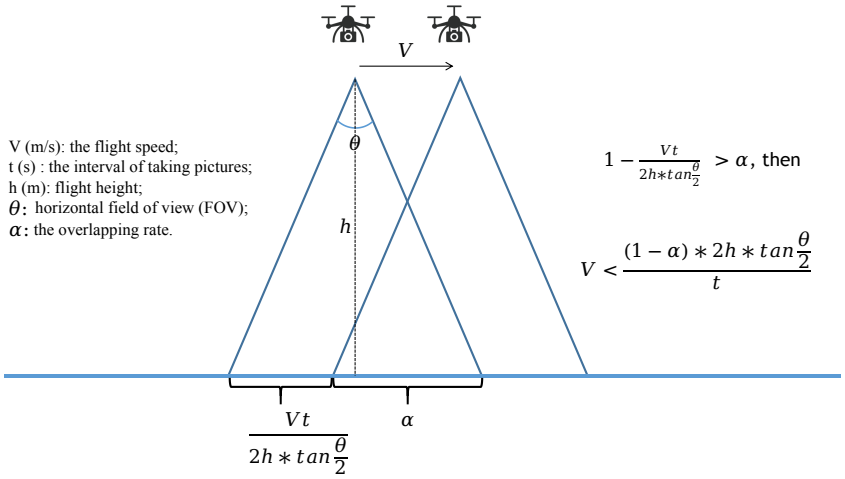
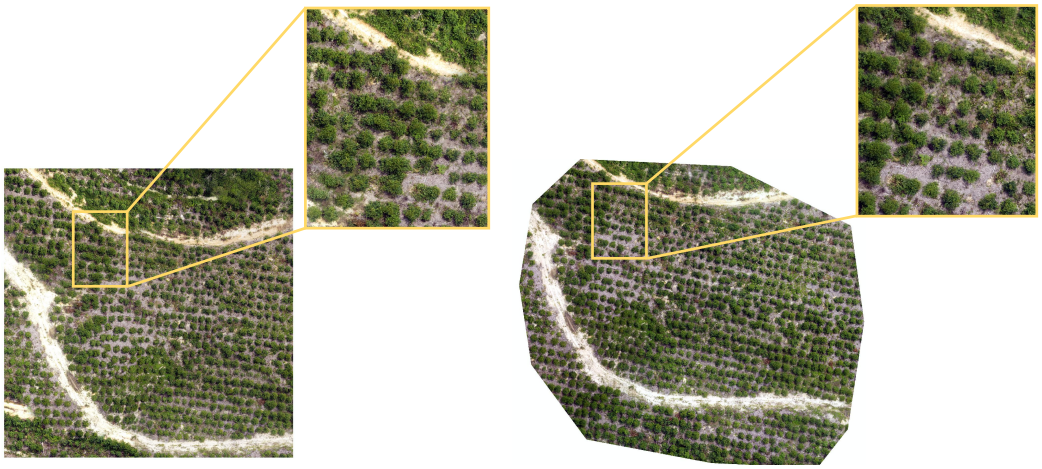


Figure 3. The relationship among the following significant parameters: flight speed is V (m/s), flight height is h (m), horizontal field of view (FOV) is θ, the overlapping rate is α, the interval of taking pictures is t (s). In order to guarantee the overlapping rate, the above-mentioned parameters should meet the equation shown in the figure.



(a) Proposed Method (b) Pix4DMapper

Figure 4. Mosaic results of the proposed method on acacia dataset. Some screenshots are highlighted to demonstrate the mosaic details. Our stitching quality is generally high compared with the offline results of state-of-the-art commercial software Pix4DMapper.

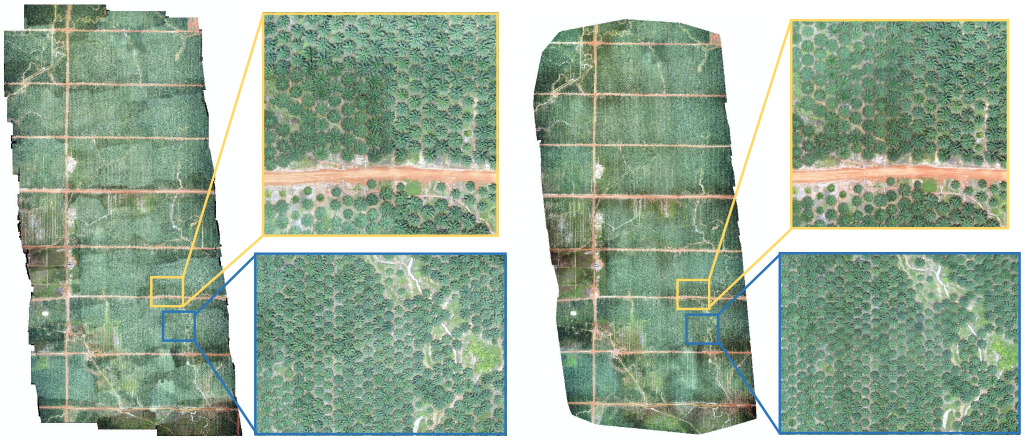


Figure 5. Orthophoto results of the proposed stitching method on the oil-palm dataset. The results show that the quality of live mosaicing is comparable to that of Pix4DMapper. Some details even show that the output live mosaic looks better than offline-based commercial construction software Pix4DMapper. The blue bounding box of Pix4DMapper shows the blur condition, which is not shown in proposed method.

In addition, our method could handle low overlapping rate conditions, but Pix4DMapper could not deal with them. The images are collected with an overlapping rate of 30% for stitching comparision experiments between the proposed method and Pix4DMapper. From the Figure 6, we could find that the proposed method is able to reconstruct the entire region whereas Pix4DMapper can only reconstruct about 1/4 of region. However, the 30% of overlapping rate is still challenging for our proposed methods, some little misaligned parts could be found in the left result of figure.

Figure 6. Images with low overlapping rate (30%) are collected for stitching comparision experiments between the proposed method (**a**) and Pix4DMapper (**b**).

4.1.2. Computation Performance Analysis

Efficiency is one of our primary concerns in our fast tree counting system. The proposed stitching method is compared with Pix4DMapper to validate the computational performance. From the Table 1, our method is much faster than Pix4DMapper with the same number and resolution images. Furthermore, compared to the offline-based method,

the proposed real-time construction system even costs less hardware resources that containing graphics processing unit (GPU) and random-access memory (RAM). Especially for oil-palm sequential images, the number of which is 189 and the area is about 2.666 square kilometers, the time cost for stitching task is just 2min19s using our proposed method, which is about 1/40 of that using offline-based Pix4DMapper. The multi-planar assumption and improved optimization method play a vital role for the stitching task.

Table 1. Time usage and peak source usage statistics for processing sequential aerial images dataset. The results show that our algorithm is much faster than other commercial software Pix4DMapper with less hardware resources usage.

Sequence	Images	Resolution	Dataset Size	Time Cost		Peak Source Usage	
				Ours	Pix4DMapper	Ours	Pix4DMapper
acacia	8	6000 × 4000	162 MB	**12 s**	4 min 20 s	**100%CPU, 57%GPU, 53%RAM**	100%CPU, 72%GPU, 74%RAM
oil-palm	189	6000 × 3376	1.8 GB	**2 min 19 s**	94 min 24 s	**100%CPU, 67%GPU, 70%RAM**	100%CPU, 80%GPU, 92%RAM

4.1.3. Discussion of the Potential Uncertainties

There are some potential uncertainties of the proposed method which need to be discussed. The yellow box of Figure 7 shows that the proposed stitching algorithm could deal with the slight light changes conditions. The SIFT descriptor is invariant to image rotation and scale changes, and has strong adaptability to 3D viewing angle changes and illumination changes. The RootSIFT algorithm [60] is adopted to generate the descriptors to further improve the matching quality and enhance the robustness of the stitching procedure. However, low overlapping rate (30%) and fast-changing conditions will increase the difficulty of the matching algorithm, which easily leads to bad stitching results like misalignment phenomenon shown in blue box of Figure 7.

Figure 7. Discussion about the influence of fast-changing light condition and low overlapping rate for our proposed stitching method. The yellow box shows the stitching result in slight light changes condition. And the blue box displays the misalignment stitching result in low overlapping rate and fast-changing condition.

4.2. Results of Point Supervised Tree Detection

4.2.1. Evaluation Metric

In the following detection experiments, mean absolute error (MAE) [66], root mean squared error (RMSE) [41], true positive rate (TPR), precision and F_1 are adopted as the evaluation metrics to validate the effectiveness of the proposed pipeline. They are defined as follows:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|e_i| \tag{11}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}|e_i|^2} \tag{12}$$

where $e_i = \hat{C}_i - C_i$, C_i and $\hat{C}_i$ are ground truth number and predicted number in the i-th image, respectively. N is the number of tiles.

$$TPR = \frac{TP}{TP + FN} \tag{13}$$

$$Prec = \frac{TP}{TP + FP} \tag{14}$$

$$F_1 = \frac{2TP}{2TP + FP + FN} \tag{15}$$

where the true positives (TP) means the true predicted tree number, false negatives (FN) represents the missing detected tree number, false positives (FP) is the false predicted tree number.

4.2.2. Merged Distance Parameter Setting

After generating the orthophoto mosaicing with sequential aerial images, the large scale orthophoto mosaicing is split into hundreds of tiles with a length and width of 256 pixels, which is effective for analysis with the http protocol and parallel computation. As shown in Figure 8, the tree in boundary would be detected in tile 1 and 2, respectively; the duplicate statistics for the boundary between the tiles will lead to inaccuracy counting statistics. If the center point distance between two adjacent tree masks is less than the preset merged distance, two points or more points would be fused into one point. Therefore, the merged distance d_α is used to filter the duplicate statistics to further correct the counting results. To evaluate the results, if the ground-truth point is among the mask regions, the predicted mask is regarded as the truth positive (TP). It could be seen that the d_α is significant to the experimental results. Thus, several pre-experiments are conducted to obtain the proper value for parameter d_α. According to the collected aerial images, we define the d_α range from 20 to 60, through several detection validation experiments, we compute precision, recall, and F_1 on acacia and oil-palm confused dataset under different merged distances d_α. From Figure 9, we could find that, 40 of d_α is the best parameter choice, which is used for the following experiments.

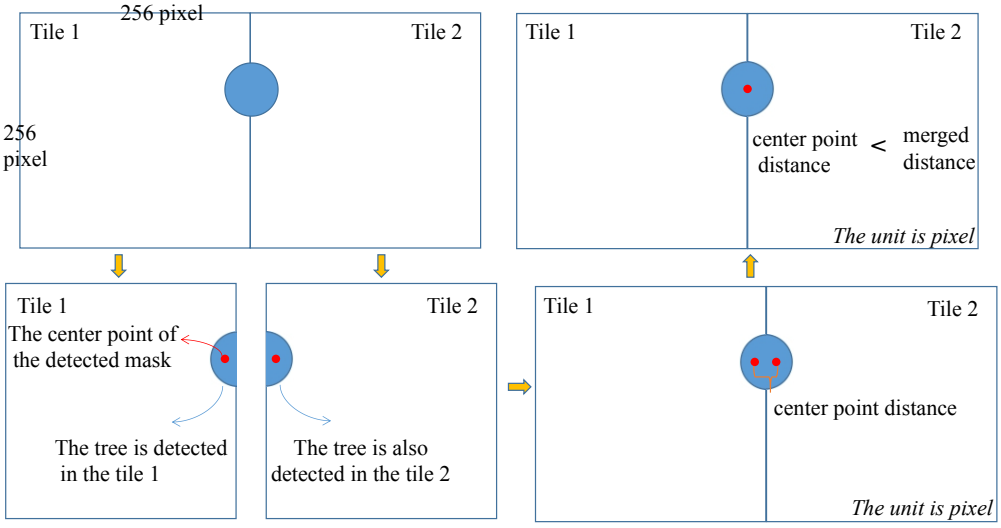


Figure 8. The description of the merged distance. If the center point distance between two adjacent tree masks is less than the preset merged distance, two points or more points in the boundaries from neighbouring tiles would be fused into one point.

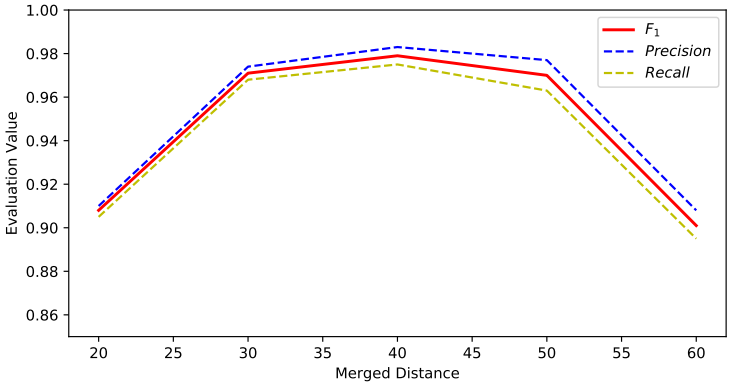


Figure 9. Precision, Recall, and F1 score under different d_α over acacia and oil-palm confused dataset. We could find that, 40 of d_α is the best parameter choice for the following experiments.

4.2.3. Comparision Experiments

For the tree counting task, the proposed framework is compared with MCNN [67], HA-CCN [68] and CAN [69], which completes tree counting task with generating density map using the regression strategy. Table 2 shows that our results achieve the best performance compared with above-mentioned methods. Good performance comes from two points: firstly, these three methods directly give the predicted number with probability density map without considering the individual's localization. Secondly, after aerial stitching for large scale areas, the mosaicing image is further split into hundreds of separated tiles. If detection results of the single tile has several biases compared with ground truth by using above mentioned MCNN [67], HA-CCN [68], and CAN [69] algorithms, bigger biases will be generated for the whole map. By the way, we could find that the counting performance of oil palm is better than that of acacia in Table 2. The main reason may come from different properties of these two type trees. Carefully review the Figures 4 and 5, it is obvious that bigger acacia trees are easily entangled with each other but this situation is not serious for oil-palm tree, which undoubtedly increases the difficulty of acacia detection with probability density map methods.

Table 2. The tree counting performance compared with MCNN, HA-CCN, and CAN on acacia and oil-palm datasets.

Methods	Acacia		Oil-Palm	
	MAE	**RMSE**	**MAE**	**RMSE**
MCNN [67]	12.32	52.06	4.48	5.53
HA-CCN [68]	4.12	18.42	3.67	4.81
CAN [69]	3.35	12.06	2.49	4.12
Ours	**2.135**	**3.274**	**2.068**	**3.159**

In order to validate the effectiveness of the proposed framework in the localization task, we conduct the comparison experiments with full supervised methods ike Faster R-CNN [42] and FPN [70] methods, and some weakly supervised methods from image level annotations, like WSDDN [71], PCL [72], and C-MIL [73]. The results are displayed in Table 3, it could be seen that the proposed point supervised detection framework is comparable to the strongly supervised detection methods and better than image level weakly supervised algorithms. These weakly methods WSDDN [71], PCL [72], and C-MIL [73] estimate the target tree localization with the feature response maps without considering the aggregation of adjacent neighboring features. In addition, our method only cost one-tenth memory compared with Faster R-CNN detection pipeline in training and validation procedures, which is a lightweight framework beneficial to be deployed in hardware devices with low calculation power.

Table 3. The tree localization performance compared with Faster R-CNN, FPN, WSDDN, PCL, and C-MIL on acacia and oil-palm datasets.

Methods	Annotation	Acacia			Oil-Palm		
		TPR	**Prec**	F_1	**TPR**	**Prec**	F_1
Faster R-CNN [42]	boundingbox	0.972	0.978	0.975	0.965	0.942	0.953
FPN [70]	boundingbox	0.974	0.976	0.976	0.979	0.988	0.984
WSDDN [71]	image level	0.702	0.776	0.715	0.736	0.758	0.9750
PCL [72]	image level	0.751	0.785	0.773	0.747	0.764	0.759
C-MIL [73]	image level	0.826	0.879	0.868	0.847	0.864	0.858
Ours	point level	**0.979**	**0.985**	**0.982**	**0.974**	**0.952**	**0.963**

4.2.4. Ablation Study

In this section, we explore the positive effects of each part of loss function in the proposed method on the acacia and oil-palm datasets. Firstly, we just use the point-loss term L_p to train the tree counting network, but the framework regards the whole image as one tree mask in Figure 10b because of unbalanced and limited point supervision information. Thus, the TPR, Precision and F_1 score are very low in Table 4.

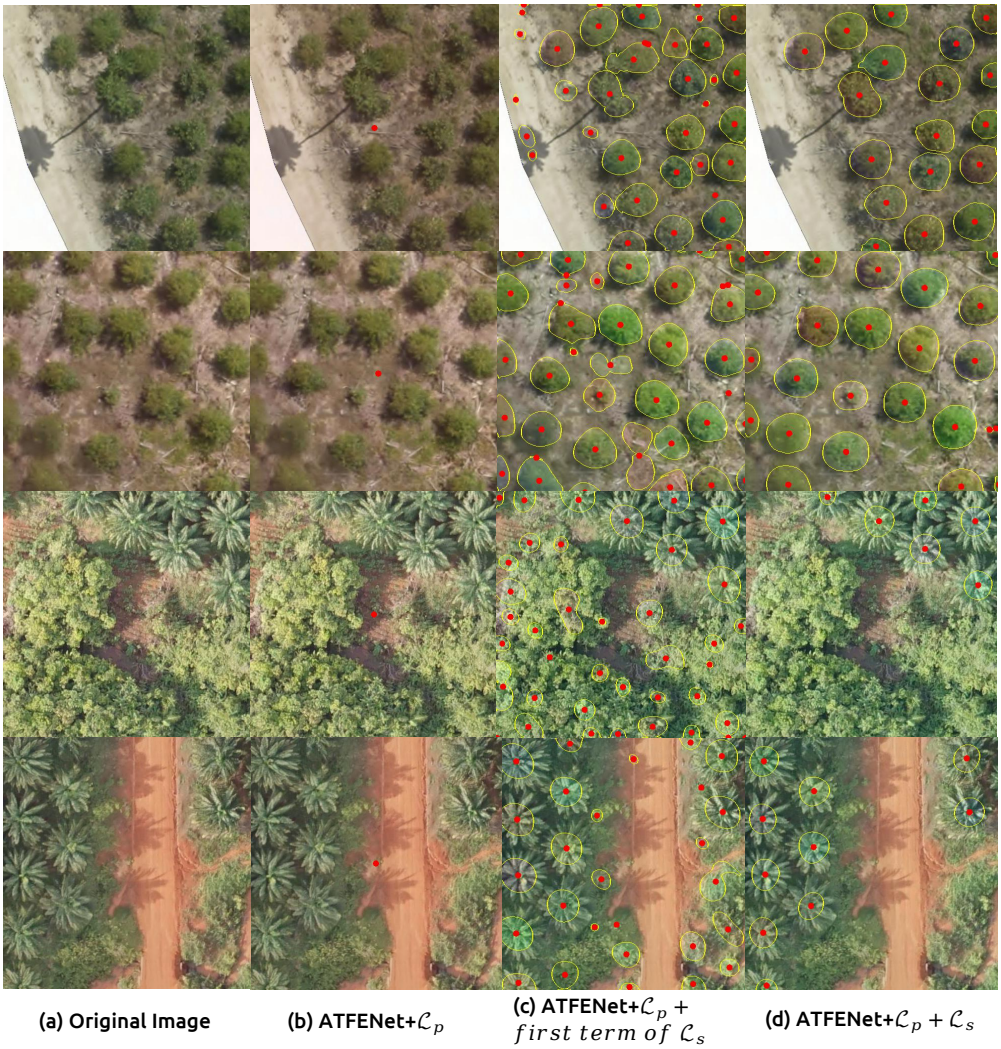


Figure 10. The qualitative results of tree counting network trained with different terms of the proposed loss function. (**a**) Test images selected from acacia and oil-palm dataset. (**b**) Inference results just using point loss. (**c**) Prediction results improved with point loss and the first term of separate loss. (**d**) The performance both using point loss and separate loss.

Therefore, the separate-loss term is introduced to encourage the network model to split the whole image into individual tree masks according to the point labels. The visualizations in Figure 10c,d show the effectiveness of this loss term. In addition, Figure 10d shows the second part of separate-loss term penalize the false positives; from the Table 4, it could be see that the proposed methods achieves good performance compared to bounding box supervised methods and other weakly supervised methods on the above-mentioned evaluation metrics.

Table 4. Ablation study of the proposed algorithm for tree localization performance.

Methods	Acacia			Oil-Palm		
	TPR	**Prec**	F_1	**TPR**	**Prec**	F_1
ATFENet +$\mathcal{L}_p$	0.062	0.147	0.087	0.075	0.141	0.098
ATFENet + $\mathcal{L}_p + \mathcal{L}_s$	**0.979**	**0.985**	**0.982**	**0.974**	**0.952**	**0.963**

4.2.5. Time Statistics

Random annotations and center click annotations. In order to evaluate the robustness of our "point in" methods and further speed up labelling, we conduct experiments with center click and random click point annotations. The random click Figure 11 is designed to obtain the point annotations that have different offsets away from the center points within the tree silhouette. From the experimental results in Table 5, we could find that the performance of random click annotations is comparable to that of center click annotations, but random click could help reduce great human labor compared with center click annotations and other full supervised annotations. In fact, the point supervised loss function could aggregate the connected image pixels into a complete mask for tree instance, and separate the connected trees with predicted mask. Therefore, the labelling time is greatly accelerated while guaranteeing the prediction accuracy.

Figure 11. The examples of random annotations on acacia and oil-palm datasets.

Table 5. The influence about tree counting performance with different annotations. Center-click and random-click annotations are conducted on acacia and oil-palm dataset.

Methods	Acacia			Oil-Palm		
	TPR	**Prec**	F_1	**TPR**	**Prec**	F_1
Center-click	0.979	0.985	0.982	0.974	0.952	0.963
Random-click	0.971	0.978	0.974	0.966	0.947	0.956

Time statistics with different annotation ways. We also consider the labeling time under different annotation ways including point, bounding box, and mask annotation in Figure 12 with labelme toolbox [62]. The annotation time cost for nearly 300 acacia trees is shown in Table 6; it is obvious that point annotation way is extremely efficient in practical applications. And the inference time of the proposed method is ten times faster than Faster R-CNN, and the speed could reach 21fps.

Table 6. The annotation time cost under different annotation ways including point, bounding box, and mask annotation.

Sequence	Number of Trees	Time Cost		
		Mask Annotation	**Bounding Box**	**Point Annotation**
acacia	nearly 300	**45 min 47 s**	**18 min 7 s**	**8 min 56 s**

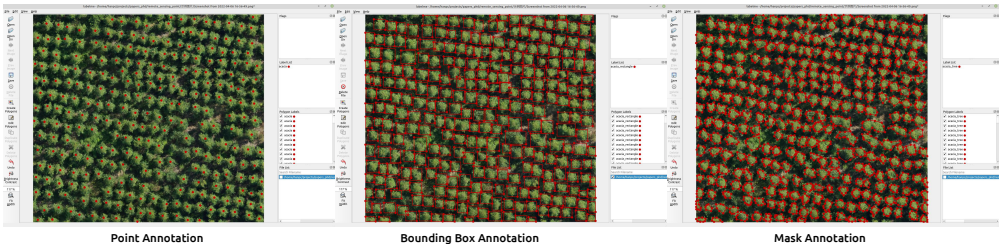


Figure 12. The annotation visualization with different annotation ways including point, bounding box, and mask annotation with the labelme toolbox.

5. Conclusions

In order to realize fast tree detection task for UAVs, we present a novel integrated vision stitching and detection framework with high efficiency and robustness. Firstly, the orthophoto mosaicing is online and real-time generated with sequential aerial images, a multiplanar hypothesis-based pose optimization method is proposed to estimate camera poses. The number of parameters about reprojection error is effectively reduced; the method could accelerate the calculation speed and achieve robust stitching performance with sequential low overlap images in the embedded devices. Secondly, the point supervised method can estimate the localization of trees and generate contour mask that is comparable to fully supervised methods. The supervised label information is easy to be obtained, which could be effective for an entire learning framework. The whole time usage by the two-stage integrated vision stitching and detection framework is much faster than traditional methods which use offline stitching method and supervised learning detection pipeline. We evaluate the proposed pipeline using sequential UAV images captured in Indonesia; the proposed pipeline achieves an F1-score of 98.2% for acacia tree detection and 96.3% for oil-palm tree detection with online orthophoto mosaicing generation. Furthermore, this two-stage vision pipeline is easily deployed into drones and the whole procedure requires no human intervention at all.

Author Contributions: Conceptualization, S.X. and S.B.; methodology, P.H.; software, P.H. and Y.Z.; validation, L.C. and J.C.; formal analysis, P.H.; investigation, L.C. and P.H.; data curation, C.Z.; writing—original draft preparation, P.H. and L.C.; writing—review and editing, P.H., S.X. and S.B.; visualization, C.Z. and T.H.; supervision, C.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China (No.41871322, 61971418, U2003109, 62171321, 62071157 and 62162044) and in part by the Shaanxi Province Key R&D Program (2022GY-253).

Data Availability Statement: The data presented in this study are available on request from the first author.

Acknowledgments: We thank Hong Kong Insight Robotics Limited for their support and assistance in data acquisition and calibration.

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

References

1. Ball, J.E.; Anderson, D.T.; Chan, C.S. Comprehensive survey of deep learning in remote sensing: Theories, tools, and challenges for the community. *J. Appl. Remote Sens.* **2017**, *11*, 042609. [CrossRef]
2. Li, W.; Fu, H.; Yu, L.; Cracknell, A. Deep learning-based oil palm tree detection and counting for high-resolution remote sensing images. *Remote Sens.* **2017**, *9*, 22. [CrossRef]
3. Li, W.; Dong, R.; Fu, H.; Yu, L. Large-scale oil palm tree detection from high-resolution satellite images using two-stage convolutional neural networks. *Remote Sens.* **2019**, *11*, 11. [CrossRef]
4. Li, W.; He, C.; Fu, H.; Zheng, J.; Dong, R.; Xia, M.; Yu, L.; Luk, W. A real-time tree crown detection approach for large-scale remote sensing images on FPGAs. *Remote Sens.* **2019**, *11*, 1025. [CrossRef]
5. Bhattacharyya, R.; Bhattacharyya, A. Crown Detection and Counting Using Satellite Images. In *Emerging Technology in Modelling and Graphics*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 765–773.
6. Zheng, J.; Fu, H.; Li, W.; Wu, W.; Yu, L.; Yuan, S.; Tao, W.Y.W.; Pang, T.K.; Kanniah, K.D. Growing status observation for oil palm trees using Unmanned Aerial Vehicle (UAV) images. *ISPRS J. Photogramm. Remote Sens.* **2021**, *173*, 95–121. [CrossRef]
7. Vallet, J.; Panissod, F.; Strecha, C.; Tracol, M. Photogrammetric Performance of an Ultra Light Weight Swinglet UAV. *ISPRS-Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2011**, *3822*, 253–258.
8. Verhoeven, G. Taking computer vision aloft—Archaeological three-dimensional reconstructions from aerial photographs with photoscan. *Archaeol. Prospect.* **2011**, *18*, 67–73. [CrossRef]
9. Zhao, Y.; Liu, G.; Xu, S.; Bu, S.; Jiang, H.; Wan, G. Fast Georeferenced Aerial Image Stitching With Absolute Rotation Averaging and Planar-Restricted Pose Graph. *IEEE Trans. Geosci. Remote. Sens.* **2020**, *59*, 3502–3517. [CrossRef]
10. Bu, S.; Zhao, Y.; Wan, G.; Liu, Z. Map2dfusion: Real-time incremental UAV image mosaicing-based on monocular slam. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016; pp. 4564–4571.
11. Botterill, T.; Mills, S.; Green, R. Real-time aerial image mosaicing. In Proceedings of the 2010 25th International Conference of Image and Vision Computing New Zealand, Queenstown, New Zealand, 8–9 November 2010; pp. 1–8.
12. de Souza, R.H.C.; Okutomi, M.; Torii, A. Real-time image mosaicing using non-rigid registration. In *Pacific-Rim Symposium on Image and Video Technology*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 311–322.
13. Schönberger, J.L.; Frahm, J.M. Structure-from-Motion Revisited. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
14. Snavely, N. Bundler: Structure from Motion (SFM) for Unordered Image Collections. 2008. Available online: http://phototour.cs.washington.edu/bundler/ (accessed on 3 January 2022).
15. Mur-Artal, R.; Montiel, J.; Tardos, J.D. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *arXiv* **2015**, arXiv:1502.00956.
16. Mur-Artal, R.; Tardós, J.D. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Trans. Robot.* **2017**, *33*, 1255–1262. [CrossRef]
17. Engel, J.; Koltun, V.; Cremers, D. Direct sparse odometry. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 611–625. [CrossRef]
18. Forster, C.; Zhang, Z.; Gassner, M.; Werlberger, M.; Scaramuzza, D. Svo: Semidirect visual odometry for monocular and multicamera systems. *IEEE Trans. Robot.* **2017**, *33*, 249–265. [CrossRef]
19. Moulon, P.; Monasse, P.; Marlet, R. Global fusion of relative motions for robust, accurate and scalable structure from motion. In Proceedings of the 2013 IEEE International Conference on Computer Vision (ICCV), Sydney, Australia, 1–8 December2013; pp. 3248–3255.
20. Sweeney, C. Theia Multiview Geometry Library: Tutorial & Reference. Available online: http://theia-sfm.org (accessed on 10 January 2022).
21. Aliero, M.M.; Mukhtar, N.; Al-Doksi, J. The usefulness of unmanned airborne vehicle (UAV) imagery for automated palm oil tree counting. *J. For. Res.* **2014**, *1*, 1–12.
22. Wang, Y.; Zhu, X.; Wu, B. Automatic detection of individual oil palm trees from UAV images using HOG features and an SVM classifier. *Int. J. Remote Sens.* **2019**, *40*, 7356–7370. [CrossRef]
23. Soula, A.; Tbarki, K.; Ksantini, R.; Said, S.B.; Lachiri, Z. A novel incremental Kernel Nonparametric SVM model (iKN-SVM) for data classification: An application to face detection. *Eng. Appl. Artif. Intell.* **2020**, *89*, 103468. [CrossRef]
24. Gao, L.; Li, J.; Khodadadzadeh, M.; Plaza, A.; Zhang, B.; He, Z.; Yan, H. Subspace-based support vector machines for hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* **2014**, *12*, 349–353.
25. Puissant, A.; Rougier, S.; Stumpf, A. Object-oriented mapping of urban trees using Random Forest classifiers. *Int. J. Appl. Earth Obs. Geoinf.* **2014**, *26*, 235–245. [CrossRef]
26. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
27. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision & Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
28. Fernández-Sanjurjo, M.; Bosquet, B.; Mucientes, M.; Brea, V.M. Real-time visual detection and tracking system for traffic monitoring. *Eng. Appl. Artif. Intell.* **2019**, *85*, 410–420. [CrossRef]
29. Tan, J. Complex object detection using deep proposal mechanism. *Eng. Appl. Artif. Intell.* **2020**, *87*, 103234. [CrossRef]

30. Kwasniewska, A.; Ruminski, J.; Szankin, M.; Kaczmarek, M. Super-resolved thermal imagery for high-accuracy facial areas detection and analysis. *Eng. Appl. Artif. Intell.* **2020**, *87*, 103263. [CrossRef]
31. Wei, X.; Yang, Z.; Liu, Y.; Wei, D.; Jia, L.; Li, Y. Railway track fastener defect detection-based on image processing and deep learning techniques: A comparative study. *Eng. Appl. Artif. Intell.* **2019**, *80*, 66–81. [CrossRef]
32. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2015, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
33. Kussul, N.; Lavreniuk, M.; Skakun, S.; Shelestov, A. Deep Learning Classification of Land Cover and Crop Types Using Remote Sensing Data. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 778–782. [CrossRef]
34. Kamilaris, A.; Prenafeta-Boldú, F.X. Deep Learning in Agriculture: A Survey. *Comput. Electron. Agric.* **2018**, *147*, 70–90. [CrossRef]
35. Liu, Q.; Xiang, X.; Wang, Y.; Luo, Z.; Fang, F. Aircraft detection in remote sensing image-based on corner clustering and deep learning. *Eng. Appl. Artif. Intell.* **2020**, *87*, 103333. [CrossRef]
36. Sarabia, R.; Aquino, A.; Ponce, J.M.; López, G.; Andújar, J.M. Automated Identification of Crop Tree Crowns From UAV Multispectral Imagery by Means of Morphological Image Analysis. *Remote Sens.* **2020**, *12*, 748. [CrossRef]
37. Zhou, Y.; Wang, L.; Jiang, K.; Xue, L.; An, F.; Chen, B.; Yun, T. Individual tree crown segmentation-based on aerial image using superpixel and topological features. *J. Appl. Remote Sens.* **2020**, *14*, 022210. [CrossRef]
38. Liu, T.; Abdelrahman, A.; Morton, J.; Wilhelm, V.L. Comparing fully convolutional networks, random forest, support vector machine, and patch-based deep convolutional neural networks for object-based wetland mapping using images from small unmanned aircraft system. *Giscience Remote Sens.* **2018**, *55*, 243–264. [CrossRef]
39. Chen, S.W.; Shivakumar, S.S.; Dcunha, S.; Das, J.; Okon, E.; Qu, C.; Taylor, C.J.; Kumar, V. Counting Apples and Oranges With Deep Learning: A Data-Driven Approach. *IEEE Robot. Autom. Lett.* **2017**, *2*, 781–788. [CrossRef]
40. Wang, Z.; Underwood, J.; Walsh, K.B. Machine vision assessment of mango orchard flowering. *Comput. Electron. Agric.* **2018**, *151*, 501–511. [CrossRef]
41. Liu, J.; Gao, C.; Meng, D.; Hauptmann, A.G. Decidenet: Counting varying density crowds through attention guided detection and density estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2018, Salt Lake City, UT, USA, 18–22 June 2018.
42. Ren, S.; Girshick, R.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef]
43. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; Springer: Cham, Switzerland, 2015; pp. 234–241.
44. Freudenberg, M.; Nölke, N.; Agostini, A.; Urban, K.; Wörgötter, F.; Kleinn, C. Large Scale Palm Tree Detection In High Resolution Satellite Images Using U-Net. *Remote. Sens.* **2019**, *11*, 312. [CrossRef]
45. Zhao, T.; Yang, Y.; Niu, H.; Wang, D.; Chen, Y. Comparing U-Net convolutional network with mask R-CNN in the performances of pomegranate tree canopy segmentation. In *Multispectral, Hyperspectral, and Ultraspectral Remote Sensing Technology, Techniques and Applications VII*; International Society for Optics and Photonics: Bellingham, WA, USA, 2018; Volume 10780, p. 107801J.
46. Papadopoulos, D.P.; Uijlings, J.R.; Keller, F.; Ferrari, V. Training object class detectors with click supervision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2017, Honolulu, HI, USA, 21–26 July 2017; pp. 6374–6383.
47. Ribera, J.; Guera, D.; Chen, Y.; Delp, E.J. Locating Objects Without Bounding Boxes. *arXiv* **2018**, arXiv:1806.07564.
48. Gherardi, R.; Farenzena, M.; Fusiello, A. Improving the efficiency of hierarchical structure-and-motion. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 1594–1600.
49. Turner, D.; Lucieer, A.; Watson, C. An automated technique for generating georectified mosaics from ultra-high resolution unmanned aerial vehicle (UAV) imagery,-based on structure from motion (SfM) point clouds. *Remote Sens.* **2012**, *4*, 1392–1410. [CrossRef]
50. Lati, A.; Belhocine, M.; Achour, N. Robust aerial image mosaicing algorithm-based on fuzzy outliers rejection. *Evol. Syst.* **2019**, *11*, 717–729. [CrossRef]
51. Guizilini, V.; Sales, D.; Lahoud, M.; Jorge, L. Embedded mosaic generation using aerial images. In Proceedings of the 2017 Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR), Curitiba, Brazil, 8–11 November 2017; pp. 1–6.
52. Itakura, K.; Hosoi, F. Automatic tree detection from three-dimensional images reconstructed from 360 spherical camera using YOLO v2. *Remote Sens.* **2020**, *12*, 988. [CrossRef]
53. Zheng, J.; Li, W.; Xia, M.; Dong, R.; Fu, H.; Yuan, S. Large-scale oil palm tree detection from high-resolution remote sensing images using faster-rcnn. In Proceedings of the IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July 2019–2 August 2019; pp. 1422–1425.
54. G Braga, J.R.; Peripato, V.; Dalagnol, R.; P Ferreira, M.; Tarabalka, Y.; OC Aragão, L.E.; F de Campos Velho, H.; Shiguemori, E.H.; Wagner, F.H. Tree crown delineation algorithm-based on a convolutional neural network. *Remote Sens.* **2020**, *12*, 1288. [CrossRef]
55. Wagner, F.H.; Dalagnol, R.; Tagle Casapia, X.; Streher, A.S.; Phillips, O.L.; Gloor, E.; Aragão, L.E. Regional mapping and spatial distribution analysis of canopy palms in an amazon forest using deep learning and VHR images. *Remote Sens.* **2020**, *12*, 2225. [CrossRef]

56. Zhang, C.; Atkinson, P.M.; George, C.; Wen, Z.; Diazgranados, M.; Gerard, F. Identifying and mapping individual plants in a highly diverse high-elevation ecosystem using UAV imagery and deep learning. *ISPRS J. Photogramm. Remote Sens.* **2020**, *169*, 280–291. [CrossRef]
57. Ferreira, M.P.; de Almeida, D.R.A.; de Almeida Papa, D.; Minervino, J.B.S.; Veras, H.F.P.; Formighieri, A.; Santos, C.A.N.; Ferreira, M.A.D.; Figueiredo, E.O.; Ferreira, E.J.L. Individual tree detection and species classification of Amazonian palms using UAV images and deep learning. *For. Ecol. Manag.* **2020**, *475*, 118397. [CrossRef]
58. Xiao, C.; Qin, R.; Huang, X. Treetop detection using convolutional neural networks trained through automatically generated pseudo labels. *Int. J. Remote Sens.* **2020**, *41*, 3010–3030. [CrossRef]
59. Brandt, M.; Tucker, C.J.; Kariryaa, A.; Rasmussen, K.; Abel, C.; Small, J.; Chave, J.; Rasmussen, L.V.; Hiernaux, P.; Diouf, A.A.; et al. An unexpectedly large count of trees in the West African Sahara and Sahel. *Nature* **2020**, *587*, 78–82. [CrossRef]
60. Khan, F.A.; Khelifi, F.; Tahir, M.A.; Bouridane, A. Dissimilarity Gaussian mixture models for efficient offline handwritten text-independent identification using SIFT and RootSIFT descriptors. *IEEE Trans. Inf. Forensics Secur.* **2018**, *14*, 289–303. [CrossRef]
61. Chatterjee, A.; Madhav Govindu, V. Efficient and robust large-scale rotation averaging. In Proceedings of the IEEE International Conference on Computer Vision 2013, Sydney, Australia, 1–8 December 2013; pp. 521–528.
62. Russell, B.C.; Torralba, A.; Murphy, K.P.; Freeman, W.T. LabelMe: A database and web-based tool for image annotation. *Int. J. Comput. Vis.* **2008**, *77*, 157–173. [CrossRef]
63. Soydaner, D. Attention mechanism in neural networks: Where it comes and where it goes. *Neural Comput. Appl.* **2022**, *34*, 13371–13385. [CrossRef]
64. Bearman, A.; Russakovsky, O.; Ferrari, V.; Fei-Fei, L. What's the point: Semantic segmentation with point supervision. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Cham, Switzerland, 2016; pp. 549–565.
65. Saarinen, K. Color image segmentation by a watershed algorithm and region adjacency graph processing. In Proceedings of the 1st International Conference on Image Processing, Austin, TX, USA, 13–16 November 1994.
66. Borji, A.; Cheng, M.M.; Jiang, H.; Li, J. Salient object detection: A benchmark. *IEEE Trans. Image Process.* **2015**, *24*, 5706–5722. [CrossRef]
67. Zhang, Y.; Zhou, D.; Chen, S.; Gao, S.; Ma, Y. Single-Image Crowd Counting via Multi-Column Convolutional Neural Network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2016, Las Vegas, NV, USA, 27–30 June 2016; pp. 589–597.
68. Sindagi, V.A.; Patel, V.M. Ha-ccn: Hierarchical attention-based crowd counting network. *IEEE Trans. Image Process.* **2019**, *29*, 323–335. [CrossRef]
69. Liu, W.; Salzmann, M.; Fua, P. Context-aware crowd counting. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5099–5108.
70. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
71. Bilen, H.; Vedaldi, A. Weakly supervised deep detection networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2846–2854.
72. Tang, P.; Wang, X.; Bai, S.; Shen, W.; Bai, X.; Liu, W.; Yuille, A. Pcl: Proposal cluster learning for weakly supervised object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *42*, 176–191. [CrossRef] [PubMed]
73. Wan, F.; Liu, C.; Ke, W.; Ji, X.; Jiao, J.; Ye, Q. C-MIL: Continuation multiple instance learning for weakly supervised object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 2199–2208.