energies

Article

Sustainable Oil Palm Resource Assessment Based on an Enhanced Deep Learning Method

Xinni Liu [1,*], Kamarul H. Ghazali [2] and Akeel A. Shah [3]

[1] School of Information, Xi'an University of Finance and Economics, Xi'an 710100, China
[2] Faculty of Electrical and Electronic Engineering Technology, Universiti Malaysia Pahang, Pekan 26600, Malaysia; kamarul@ump.edu.my
[3] Key Laboratory of Low-Grade Energy Utilization Technologies and Systems, MOE, Chongqing University, Chongqing 400030, China; akeelshah@cqu.edu.cn
* Correspondence: lxinni@163.com

Abstract: Knowledge of the number and distribution of oil palm trees during the crop cycle is vital for sustainable management and predicting yields. The accuracy of the conventional image processing method is limited for the hand-crafted feature extraction method and the overfitting problem occurs due to the insufficient dataset. We propose a modification of the Faster Region-based Convolutional Neural Network (FRCNN) for palm tree detection to reduce the overfitting problem and improve the detection accuracy. The enhanced FRCNN (EFRCNN) leads to improved performance for detecting objects (in the same image) when they are of multiple sizes by using a feature concatenation method. Transfer learning based on a ResNet50 model is used to extract the features of the input image. High-resolution images of oil palm trees from a drone are used to form the data set, containing mature, young, and mixed oil palm tree regions. We train and test the EFRCNN, the FRCNN, a CNN used recently for oil palm image detection, and two standard methods, namely, the support vector machine (SVM) and template matching (TM). The results reveal an overall accuracy of ≥96.8% for the EFRCNN on the three test sets. The accuracy is higher than the CNN and FRCNN and substantially higher than SVM and TM. For large-scale plantations, the accuracy improvement is significant. This research provides a method for automatically counting the oil palm trees in large-scale plantations.

Keywords: sustainable; oil palm tree; resource assessment; deep learning; Faster Region-Based Convolutional Neural Network; feature map concatenation

check for updates

Citation: Liu, X.; Ghazali, K.H.; Shah, A.A. Sustainable Oil Palm Resource Assessment Based on an Enhanced Deep Learning Method. *Energies* **2022**, *15*, 4479. https://doi.org/10.3390/en15124479

Academic Editor: Attilio Converti

Received: 29 April 2022
Accepted: 17 June 2022
Published: 20 June 2022

1. Introduction

The oil palm tree originated from West Africanis and was brought by the British to Malaysia in the early 1870s [1]. In modern times, the oil palm tree is a commodity crop in some tropical regions such as Indonesia and Malaysia [2]. With 5.85 million hectares of oil palm tree plantation, Malaysia has become the world's major producer and exporter of palm oil (the most widely used vegetable oil in the world), taking up more than 60% of the agricultural land [3]. This makes the oil palm industry one of the most important contributors to Malaysia's GDP (Gross Domestic Product).

Precision farming is a farming management system that uses modern technologies in the crop production process to help understand and efficiently manage farms [4]. It involves: measuring and analyzing variability in yield, solid quality, pests, and weeds; decision-making; differential actions; and the assessment of outcomes. Implementation of precision farming by the plantation company ensures optimum productivity, quality, and economic return, and also helps to mitigate environmental impacts. Traditional methods of acquisition for images of oil palm tree plantations use remote sensing. The satellite image can be obtained either from high spatial resolution data associated with an airborne imaging spectrometer or from a high spatial resolution satellite, such as the QuickBird [5]. Recently, with the development of small multispectral and hyperspectral imaging sensors, drone images are becoming increasingly popular for the acquisition of high-resolution and spectral measurements [6].

The task of measuring and analyzing variability, which involves some form of spatio-temporal mapping, is perhaps the most important aspect of precision farming [7]. Spatial and plotting map data need to be retrieved quickly and translated into knowledge that can help to improve production. This can involve manual counting, the use of crop models, machine learning, and data mining tools. In the standard method, the trees in an oil palm tree plantation are counted manually from the acquired image. A Geographic Information System (GIS) software is used, which is tedious and inefficient for large-scale plantations.

More recently, machine learning has been proven to be an effective and flexible tool in machine vision agriculture systems for analyzing images and decision-making [8]. The machine learning method usually extracts features from the image first, and then classifies the extracted features to classify the various objects in the image [9]. However, the feature extraction approach is usually hand-crafted and the features are designed manually by human experts a priori to extract a set of chosen characteristics. Manandhar et al. used shape features by extracting the polar shape matrix from a remote sensing image of an oil palm tree plantation; a local maximum detection algorithm was used to detect the objects in eight different oil palm tree images [10]. Malek et al. used the popular scale-invariant feature transform (SIFT) and a classifier to detect palm trees from unmanned aerial vehicle images [11]. Therefore, the features can be difficult to design and are specific to a particular data set.

An alternative is learned features, a machine learning approach that is used to automatically discover the features for classification using the raw data. In particular, deep learning algorithms can extract and discriminate between high-level features from images and have established themselves as the preferred learning approach for such applications [12]. In recent years, deep learning-based image classification algorithms have successfully been used in oil palm tree detection from aerial imagery [13–15]. However, most of the current deep learning methods focus on remote sensing imagery and the methods do not cover the case in which multiple-sized oil palm trees are present in the images. Li et al. used a two-stage method, in which they first trained a Convolutional Neural Network (CNN), and then used the sliding window method to obtain the final detection result. This two-stage method usually requires a large training dataset and the sliding window method also makes it time-consuming [13]. Mubin et al. trained two CNNs to detect images with young and mature oil palm trees. A real oil palm tree plantation, on the other hand, contains trees of multiple sizes, and thus, a model that can detect multiple sizes would be hugely beneficial [15].

The Region-based CNN (R-CNN) uses a selective search (SS) to propose a large number of regions of interest (ROI), which are fed into a CNN to extract feature vectors for classification. The fast R-CNN instead takes the image and region proposals as inputs in a CNN architecture with a single forward propagation, combining the CNN, ROI pooling and classification in one complete architecture. This method is still time-consuming because it needs to perform a SS. The faster R-CNN (FRCNN) overcomes these deficiencies by using a feature extraction network (pretrained CNN), followed by a network to generate object proposals (Regional Proposal Network (RPN)), and finally, a classification layer [16]. In this manner, the SS is eliminated. The faster R-CNN has been used successfully for vehicle detection, banana tree detection, and building detection from remote sensing images [17–19].

In order to develop an automatic oil palm tree detection approach that can detect and count multiple size oil palm trees, we introduce an enhanced FRCNN (EFRCNN) that uses feature concatenation and transfer learning from a Residual Network (ResNet) [20]. It modifies the basic FRCNN by using a feature concatenation method which integrates low-level and high-level features from a pretrained ResNet50 to increase the accuracy of detection. High-level detail obtained from the final convolution block is ideal for detecting large objects, whereas low-level information obtained from the preceding convolution blocks is ideal for detecting small objects. Combining the different levels through concatenation can improve detection if objects of different sizes are present in the image. To test the method, we generate high-resolution RGB images of a plantation using a multirotor drone. The results are compared with an SVM (with separate feature extraction), template matching (TM), and the recent methods of Mubin et al. [15] and Liu. et al. [21], the latter of which is the original FRCNN. The EFRCNN is far superior to the SVM and TM, while exhibiting small but noticeable improvements compared to the CNN method, which suggests significant advantages for large plantations.

2. Materials and Methods

2.1. Dataset

For the deep learning method, it is important to have sufficient data to avoid overfitting and ensure good generalization for detection and classification. Currently, there is no existing public dataset for oil palm tree plantations. Due to decreasing prices, coupled with technological developments, drones have become more popular and widely used in modern agriculture for machine vision applications. Multirotor drones are the most common types of drones used by professionals for applications such as aerial photography [22].

Therefore, in this study, a multirotor drone was used to collect the data from an oil palm tree plantation. The drone used was a LiAir 220, which is an UAV-mounted system developed by GreenValley International (Figure 1). The LiAir 220 is equipped with a 40-channel Pandar40 laser sensor (Hesai, Shanghai) with a 220 m range. The range accuracy is 2 cm, the scan rate is 700,000 pts/s, and the camera used is a Sony a6000 with 24 megapixels. The drone was flown above the plantation and captured high-resolution RGB images of each block. The images were then spliced to form an integral picture of the whole plantation.

Figure 1. The LiAir 220 drone used for high-resolution image capture of an oil palm tree planation.

Figure 2 shows the oil palm tree plantation image cropped from the collected high-resolution data. Since the dataset plays a vital role in training and the high-resolution image cannot be processed in one pass due to computational limitations, in this study, it was divided into multiple sub-images. A total of 360 sub-images with a resolution of 500 × 600 pixels were created. In turn, these images were divided into a training dataset with 260 images and a test dataset with 100 images. To ensure that the dataset contains sufficient information, an image processing method was used for augmentation.

Figure 3 shows three samples from the training dataset. The left-hand column shows the original sub-images and right-hand column shows the corresponding enhanced images. One hundred sixty of the images were randomly selected and enhanced by improving the brightness and contrast. It can be seen that the object is easier to identify from the enhanced images. Both the original and enhanced images were used as training data. After the data augmentation, the training dataset therefore contained 420 images. Subsequently, the images in the training dataset were labelled using LabelImg (Available opensource on GitHub: https://tzutalin.github.io/labelImg/ (accessed on 5 January 2020)). There are only two classes in this study, namely, oil palm tree and background.

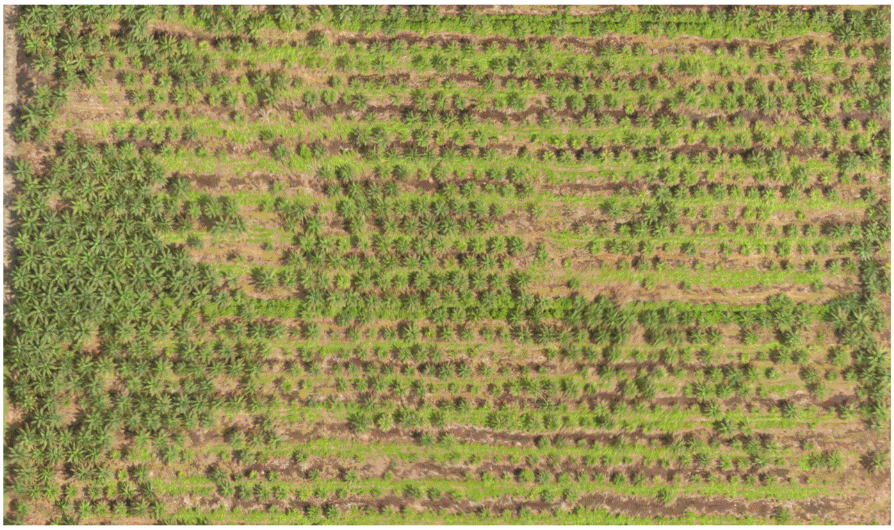

Figure 2. Oil palm tree plantation image from a drone.

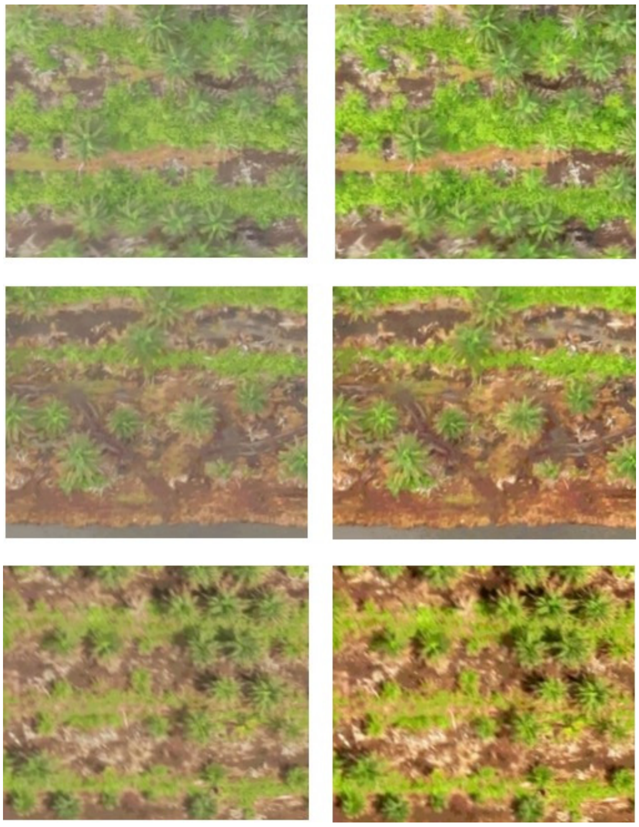

Figure 3. Original oil palm tree images (left-hand column) and enhanced images (right-hand column).

2.2. Faster RCNN and Enhanced Faster RCNN

The first step of the Faster RCNN approach passes the entire image through a pre-trained CNN that returns feature maps for the image. The conventional CNN has multiple layers with a large number of weights. Numerical experiments have revealed that network depth is of crucial importance for improving performance [23]. Experiments on the ImageNet dataset all use very deep models with at least sixteen layers. A large number of visual recognition tasks also leverage these very deep models [24]. However, as the depth of the network increases, the number of network weights also increases, which necessitates a larger training dataset. Training a model from scratch can be very computationally intensive. For classification tasks, a high-level (low-level) feature is suitable for targets of large (small) size [25]. If the image contains both small- and large-sized objects, employing a high-level feature layer is not optimal, especially in this study in which the image contains mature and very young palm trees (i.e., multiple sizes).

Based on the above considerations, in this work, a deep CNN network is employed for the accurate detection. A 50-layer Residual Network (ResNet50) is used, together with feature concatenation and transfer learning to improve the performance of the Faster RCNN. The ResNet50 adds skip/residual connections in stacked residual blocks, which leads to quick convergence and therefore, faster training by avoiding the vanishing gradient issue. This was motivated by the observation that with an increase in network depth, accuracy becomes saturated before degrading rapidly [26]. He et al. hypothesized that it should be easier for the network to learn perturbations from an identity mapping than to learn the mapping itself [20]. This led to a residual formulation across a number of so-called residual blocks of at least two layers: $y = F(x, \{w_i\}) + x$, in which x and y are the input to and output from the block, $\{w_i\}$ is the set of weights across the block (Figure 4 illustrates a block with two layers). $F(x, \{w_i\})$ represents the action of the layers in the block and is the residual mapping to be learned by the block, i.e., $F(x, \{w_i\}) = y - x$. The addition is elementwise, provided $F(x, \{w_i\})$ and x are in the same space, otherwise the input is linearly projected onto the space in which $F(x, \{w_i\})$ resides.

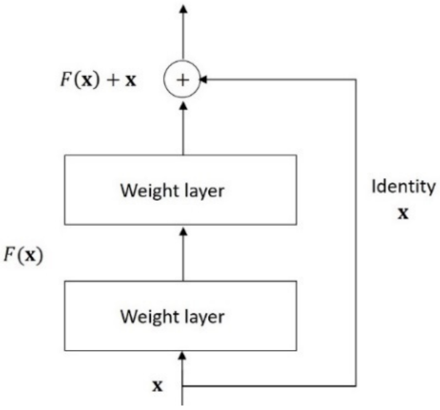

Figure 4. Illustration of a residual block in ResNet.

Feature concatenation is used to integrate the low-level and high-level features to increase the accuracy of detection. Moreover, transfer learning is employed to reduce the training cost of a new network by using a pretrained ResNet50. This leads to an enhanced Faster RCNN (which we call EFRCNN), as illustrated in Figure 5. The input is the cropped oil palm tree image and the output is the class probability and the location of the object.

The ResNet50 is pretrained on the openly available ImageNet data set. The pretrained ResNet only retains the weights of Conv1 and all the residual blocks (Conv2, Conv3, Conv4, and Conv5), discarding weights from the other layers. The extracted feature map is fed into the RPN to obtain the proposal boxes. Subsequently, the ROI pooling layer utilizes the transferred feature maps at different levels (conv3_3, conv4_3, and conv5_3) and the proposal boxes to extract the feature map for the proposal boxes. The results are then concatenated and fed into a full connected (FC) layer and then the layers are classified and refined to yield the final detection result and calculate the loss. Finally, the backpropagation (BP) algorithm is used to adjust the weights for the RPN, FC layers, classification layer, and refine the bounding box layer.

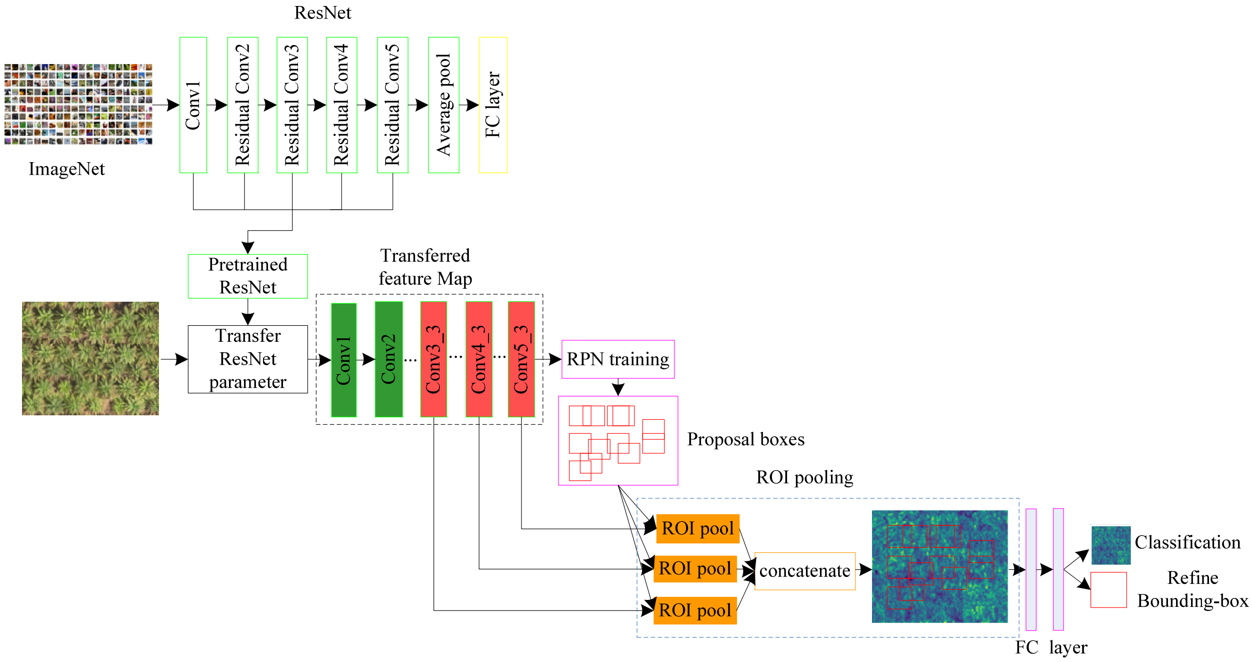

Figure 5. Pipeline of oil palm tree detection based on EFRCNN.

The RPN is a fully convolutional network that takes an image of any size as the input, while outputting a set of object proposals (rectangular), attached to each of which is an objectness score. The RPN and Fast R-CNN component share a common set of convolutional layers; region proposals are generated by sliding a small network over the feature map that is output by the final convolutional layer that is shared between the networks [16]. The input to this small network is a spatial window of the input feature map, which is mapped to a feature in a lower-dimensional space [27]. This reduced size feature is input to two fully connected convolutional layers, namely, the box regression layer (*regL*) and the softmax classification layer (*clsL*).

At each location of the sliding window, multiple region proposals are predicted, with a defined maximum of k proposals (parameterized with respect to reference boxes, referred to as anchors). The regression layer therefore returns $4k$ outputs and the classification layer returns $2k$ outputs, representing the probabilities of each of the two classes. The anchors are centered at the sliding window and associated to each are three scales and three aspect ratios (in the default scheme), which yields a total of $k = 9$ anchors for each sliding position.

The loss function for training the EFRCNN contains two parts (Equation (1)), including the classification loss L_{cls} (binary cross-entropy) and the region loss L_{reg} :

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \mu \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*) \quad (1)$$

in which $\{\cdot\}$ denotes a set. The training objective for the EFRCNN is to minimize the loss. In Equation (1), i is the index of the bounding box; p_i is the foreground softmax probability and p_i^* is the probability for the ground truth box; t_i is a vector of parameterized coordinates of the predicted bounding box and t_i^* are the coordinates of the ground truth box. The two terms are normalized and a parameter μ is used to balance the contribution of each term. The classification term is normalized by the number of classes $N_{cls} = 2$, while the region loss is normalized by the number of region proposals. A value of $N_{reg} = 300$ was used in this study. In the results presented in the next section, a value of $\mu = 2$ was adopted.

Parameterizations of the four coordinates in the bounding box regression layer are as follows [28]:

$$t_x = \frac{(x - x_a)}{w_a} \;\; t_\mathrm{y} = \frac{(y - y_a)}{h_a} \tag{2}$$

$$t_w = \log\frac{w}{w_a} \;\; t_h = \log\frac{h}{h_a} \tag{3}$$

$$t_x^* = \frac{(x^* - x_a)}{w_a} \;\; t_y^* = \frac{(y^* - y_a)}{h_a} \tag{4}$$

$$t_w^* = \log\frac{w^*}{w_a} \;\; t_h^* = \log\frac{h^*}{h_a} \tag{5}$$

In which x, y, w, and h denote the center coordinates, width, and height of the bounding box, respectively. The subscript a and superscript * denote the prediction and ground truth values, respectively.

Training is performed with backpropagation (BP) using a stochastic gradient descent (SGD) for efficiency [29]. Let $f(x)$ be the loss function and $f_i(x)$ be the loss function corresponding to each of the n training samples. Then,

$$\nabla f(x) = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(x) \tag{6}$$

SGD reduces computational cost at each iteration of BP from $O(n)$ to $O(1)$ by uniformly sampling an index $i \in \{1, \ldots, n\}$ and computing the gradient alone $\nabla f_i(x)$ to update x according to the update rule,

$$x \leftarrow x - \eta \nabla f_i(x) \tag{7}$$

in which η is the learning rate. Normally, momentum is added to reduce oscillatory behavior and promote faster convergence:

$$x \leftarrow x - \alpha \mathrm{u}_{i-1} - \eta \nabla f_i(x) \tag{8}$$

in which α is a constant and u_{i-1} is the update at the previous iteration. SGD can also be performed with mini batches, meaning a subset $I \subseteq \{1, \ldots, n\}$ is chosen randomly and the update is based on in which $|I|$ is the cardinality of I.

$$\frac{1}{|I|}\sum_{i \in I} \nabla f_i(x) \tag{9}$$

Convolutional layers are initialized using a pretrained network, as mentioned earlier. Weights in the other layers are initialized by sampling from a zero-mean Gaussian distribution with a standard deviation of 0.01. The learning rate was set at 0.001 and a momentum of 0.9 was used alongside a weight decay of 0.0005.

Algorithm 1 shows the pseudocode for EFRCNN. The number of epochs can be adjusted. The iteration in j indicates the number of the training batch in each epoch and the total number is determined by the number of training images and the batch size. Different batch sizes were attempted and larger batch sizes gave superior performance. In the results presented below, a batch size of 420 was used.

During training, the weights for the ResNet are fixed since it is trained on ImageNet. The training images are input to the pretrained ResNet to provide the feature maps $fmp3\_3$, $fmp4\_3$, and $fmp4 = 5\_3$. Then, $fmp5\_3$ is fed into the RPN network for training to obtain the proposal box, pbox, and the weights for the RPN network are updated accordingly. The maps $fmp3\_3$, $fmp4\_3$, $fmp5\_3$, and pbox are fed into the ROI network to extract the feature maps and the proposal box, respectively. The maps and bounding box are then concatenated to obtain the final feature map named $pbox\_fmap$, which will be input to the fully connected layers.

Algorithm 1. Enhanced Faster RCNN (EFRCNN)

```
Input: (epoch, iterations, image)
Output: trained EFRCNN
procedure EFRCNN(epoch,image)
    for i = 1 to epoch do
        for j = 1 to iterations do
            fmp3_3, fmp4_3, fmp5_3 = pretrained_ResNet(training image)
            pbox = transfer_learning(fmp5_3)
            update weights_RPN
            pbox_fmap= transfer_learning(fmp3_3, fmp4_3, fmp5_3, pbox)
            transfer_learning of classification and bounding box regression layers)
            update weights_FC, weights_regL, weights_clsL
            calculate loss function Loss
            update weights using BP algorithm
        end for
        if Loss<criterion
            save weights of EFRCNN w
    end for
    return w
end procedure
```

The softmax classifier and bounding box regression layers are used to conduct the final classification and regression. After training on all of the images, the loss function is used to compute the loss in the classification and box location and subsequently, the BP algorithm is used to update the weights of the FC layer, the softmax classifier, and the bounding box regression layer. The weights of the EFRCNN will be saved if the loss meets the criterion. The training process runs until it reaches the max epoch or the loss fails to further improve, at which point the weights are saved. The trained model is tested on the test dataset.

3. Results and Discussion

The proposed EFRCNN was evaluated on the test dataset and another three high-resolution images. For the test dataset, the EFRCNN model was first trained on the training dataset and then used to detect each image in the test set. For the high-resolution image, the detection process is illustrated in Figure 6. Firstly, the image is divided into sub-images, with some being the same size as the training image and others being smaller than the training image. The trained EFRCNN is then used to detect the trees in the sub-image. Finally, the detection results for all the sub-images are combined to yield the final result. The EFRCNN required approximately 1200 epochs for training.

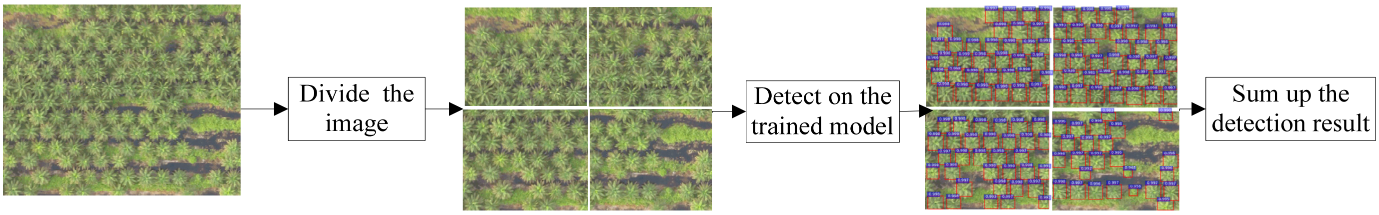

Figure 6. Workflow for the detection of the high-resolution images.

In order to evaluate the performance of the proposed EFRCNN approach, the results are compared to the LeNet CNN method used by Mubin et al. [15], the traditional support vector machine (SVM) with a linear kernel and the template match (TM) method [30,31], and the original FRCNN method [21]. The CNN LeNet contains four convolutional layers with kernel size 5 × 5, max-pooling layers with a pool size of 2 × 2 and dropout layers with a rate of 0.5. The ReLU activation function is used. For training, the Adaptive Moment Estimation (Adam) method was used. The mini batch size is 20 during the training. The training image has a size of 80 × 80 pixels, which contains two categories called oil palm tree and background. The training dataset includes 1930 positive samples with an oil palm tree in the image and 2062 negative samples.

For the SVM, part of the original image is cropped into small sub-images of 80 × 80 pixels because a single mature palm tree occupies at most 80 × 80 pixels. The cropped images are classified as either oil palm tree or background. A total of 3883 samples were cropped, including 1890 positive samples with an oil palm tree in the image and 1993 negative samples with background. These samples formed the training dataset. Figure 7 shows the training samples of oil palm trees and backgrround.

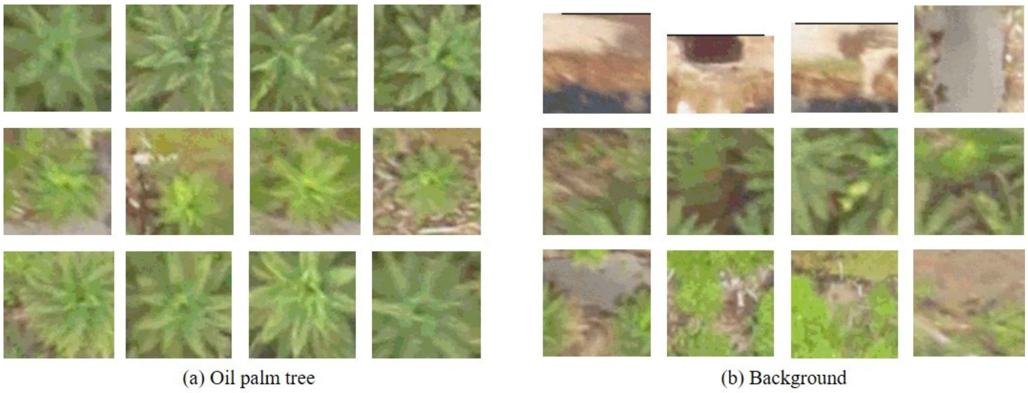

Figure 7. Training samples for SVM and TM.

For the TM method, the 1890 positive samples were used as the template dataset, along with a sliding window of 40 × 40 pixels. The CV_TM_SQDIFF_NORMED from OpenCV (https://opencv.org/ (accessed on 6 March 2019)) was used as the matching approach, which calculates the sum of the difference between the intensities in the sliding window and the template data at each pixel, and normalizes by the product of the sum of squares of the intensities in the window and template. After performing TM, the proposals were filtered using non-maximum suppression (NMS) to obtain the final detection result [32].

In this study, we use precision, recall, and overall accuracy (OA) to evaluate the performance of the different methods:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{10}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{11}$$

$$\text{OA} = \frac{\text{Precision} + \text{Recall}}{2} \tag{12}$$

in which *TP* denotes the number of oil palm trees that were correctly detected, *FP* is the number incorrectly detected, and *FN* is the number not detected.

Table 1 gives the detection accuracies of the EFRCNN, CNN, SVM, TM, and FRCNN on the test dataset. The oil palm trees in the testing dataset are manually counted and there are 2640 oil palm trees in total. In this study, the manually counted results are considered as the ground truth. The proposed model detected 2650 oil palm trees. Of these, 2582 were correctly detected, while 68 were false positives, and 58 palm trees were not detected.

Table 1. Accuracy comparison on the testing dataset.

Method	TP	FP	FN	Precision	Recall	Overall Accuracy
SVM	1990	350	650	85.0%	75.6%	80.3%
TM	1852	386	788	82.8%	70.2%	76.5%
CNN	2548	86	92	96.7%	96.5%	96.6%
FRCNN	2554	76	86	97.1%	96.7%	96.9%
EFRCNN	2582	68	58	97.4%	97.8%	97.6%

The precision, recall, and overall accuracy using the EFRCNN on the test dataset were 97.4%, 97.8%, and 97.6%, respectively, which shows the proposed model achieves a very high accuracy. The comparison in Table 1 shows that the proposed EFRCNN model outperforms SVM and TM by a wide margin in terms of both precision and recall; the overall accuracy is more than 17 percentage points higher than the next best method. The overall accuracy for CNN at 96.6% is also high, but 1.0% less than EFRCNN. The overall accuracy for FRCNN is at 96.9%, but 0.7% less than EFRCNN.

Table 2 shows the detection accuracy comparison on the region with mature oil palm trees in Figure 8a. It can be seen that the EFRCNN achieves the highest precision and it outperforms the other methods with an overall accuracy at 96.9%, while the overall accuracy for the traditional SVM and TM methods are less than 83%. Table 3 shows the accuracy comparison on the region with young oil palm trees in Figure 8b, where the overall accuracy for the proposed model is more than 96%, while the traditional methods are less than 75% in overall accuracy. The performance of these methods in terms of precision and recall is particularly poor. The performance of the CNN and FRCNN also remains high in all of the performance measures, although they are slightly lower than that of EFRCNN.

Table 2. Accuracy comparison on mature palm tree region.

Method	TP	FP	FN	Precision	Recall	Overall Accuracy
SVM	464	50	154	90.3%	75.1%	82.7%
TM	446	84	172	84.2%	72.2%	78.2%
CNN	589	24	29	96.1%	95.3%	95.7%
FRCNN	590	20	28	96.7%	95.5%	96.1%
EFRCNN	595	15	23	97.5%	96.3%	96.9%

Table 3. Accuracy comparison on young palm tree region.

Method	TP	FP	FN	Precision	Recall	Overall Accuracy
SVM	396	58	237	87.2%	62.6%	74.9%
TM	426	92	207	82.2%	67.3%	74.8%
CNN	598	25	35	96.0%	94.5%	95.3%
FRCNN	604	23	29	96.3%	95.4%	95.9%
EFRCNN	611	18	22	97.1%	96.5%	96.8%

The methods were also evaluated on three regions with different sizes of oil palm trees: mature palm trees, young palm trees, and mixed-age palm trees. The images have a size of 3600 × 2000 pixels, cropped from the original high-resolution drone image. For the EFRCNN, the high-resolution image is split into sub-images with a resolution of 500 × 600, which is the same size as the training data. The trained EFRCNN model was used to detect oil palm trees in each image.

Table 4 shows the accuracy comparison on the mixed oil palm trees image shown in Figure 8c. These numbers are consistent with the preceding results, in that EFRCNN, FRCNN, and CNN are far superior to the SVM and TM, with the EFRCNN exhibiting a slightly better performance over FRCNN and CNN.

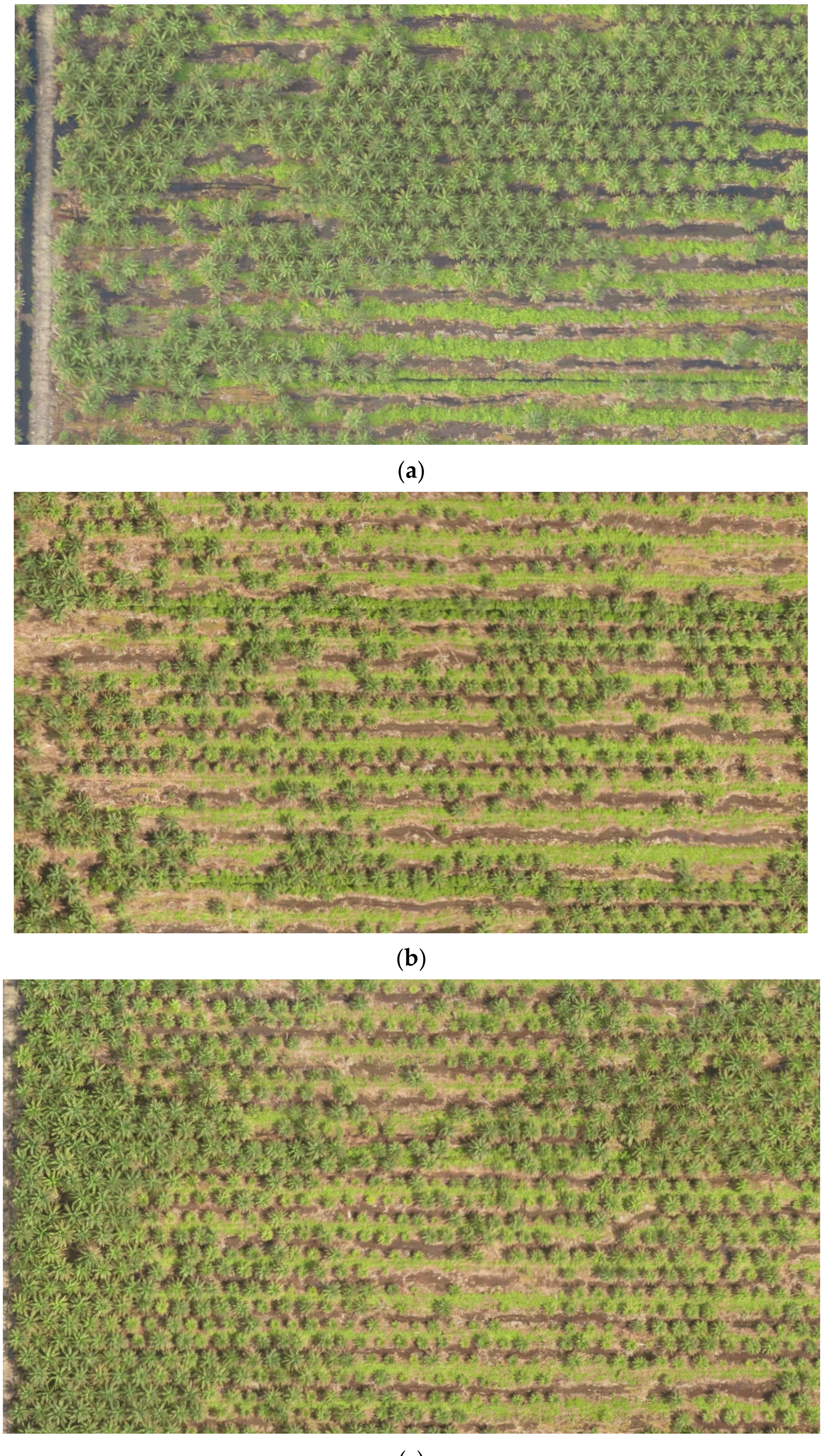

Figure 8. Three type of oil palm tree regions. (**a**) Region with mature palm trees. (**b**) Region with young palm trees. (**c**) Region with mixed palm trees.

The detection results in Table 4 shows that the overall accuracy for the proposed EFRCNN in the mixed palm tree image is slightly higher than those for the mature and young palm tree images. The sizes and distributions of the oil palm trees are different in the three test images. As seen in Figure 8a containing mature oil palm trees, some trees are overlapping with other vegetation. In Figure 8b showing young palm trees, in some regions, the trees are distributed sparsely. The detection results show that even under these challenging conditions, our proposed method and the CNN and FRCNN perform extremely well, although the EFRCNN has a 1.5% and 0.9% higher overall accuracy, respectively.

Table 4. Accuracy comparison on mixed palm tree region.

Method	TP	FP	FN	Precision	Recall	Overall Accuracy
SVM	568	85	298	87.0%	65.6%	76.3%
TM	594	126	272	82.5%	68.6%	75.6%
CNN	836	43	30	95.1%	96.5%	95.8%
FRCNN	823	22	43	97.4%	95.0%	96.2%
EFRCNN	833	16	33	98.1%	96.2%	97.2%

Figure 9 shows the detection results for several kinds of palm tree images. The red box indicates that the oil palm tree is detected correctly with a confidence score. The blue box indicates that the palm tree is incorrectly detected, which shows that the other vegetation is detected as an oil palm tree. The yellow box indicates that the palm tree is not detected. The detection results show that all the mature palm trees and young palm trees were correctly detected. The mature palm trees' confidence score is more than 0.99, while the young palm trees' confidence score is more than 0.59. One young oil palm tree was not detected in the mixed palm tree images and other vegetation was detected as oil palm trees in Figure 9d. The detection results of the samples show that most of the oil palm trees can be correctly detected.

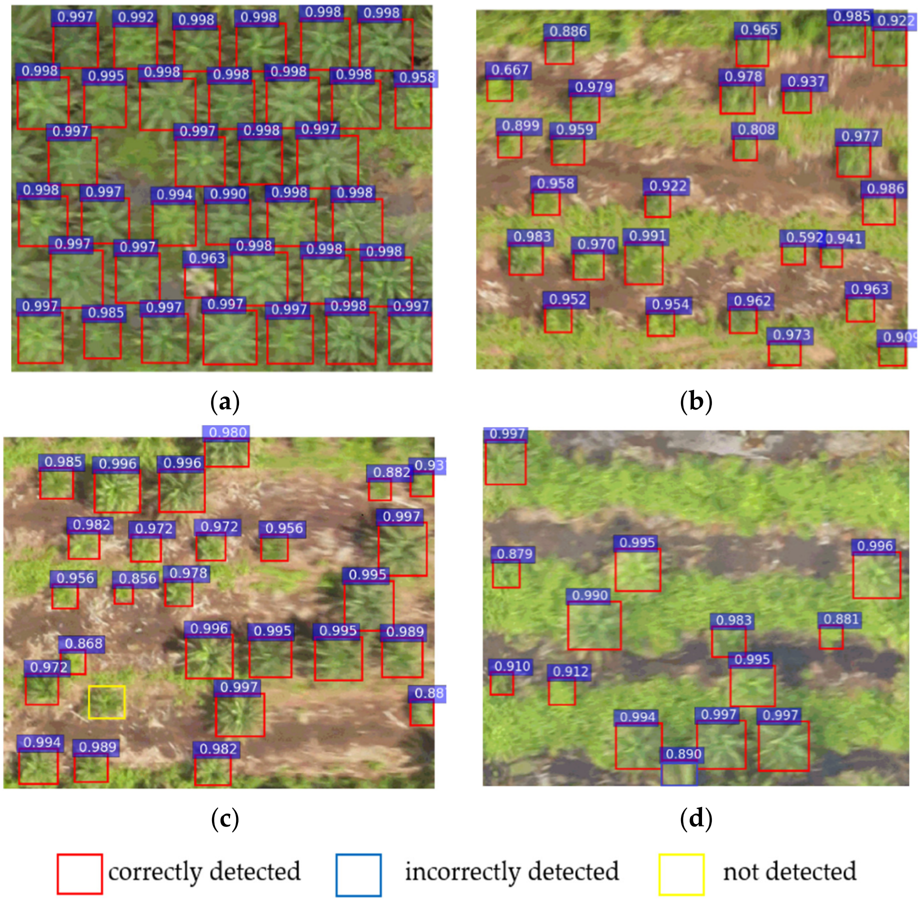

Figure 9. Detection results of palm tree images. (**a**) Mature palm trees. (**b**) Young palm trees. (**c**) Mixed palm trees. (**d**) Overlap with vegetation.

4. Conclusions

The oil palm industry is important for the development of agriculture in Malaysia. Precision farming can effectively improve the efficiency of the crop product. This study proposes a deep learning method based on an Enhanced Faster RCNN model for automatically detecting and counting oil palm trees from drone images. The original Faster RCNN is improved by using a high-level and low-level feature concatenation approach so that objects of different sizes can be better detected. The performance improvement compared to the classical SVM and template matching methods is substantial. The latter two methods are shown to be inadequate for the present application. The accuracy compared to the CNN and FRCNN is small but nontrivial (on the order of 25 more trees correctly counted in this study). For large-scale plantations, the proposed method would automatically assess the number of trees and it would present a significant advantage for sustainable oil palm resource assessment.

Author Contributions: X.L. conceived and wrote the paper; X.L. and A.A.S. analyzed the data; X.L. and K.H.G. designed the experiments and analyzed the experimental data; K.H.G. proposed the theory. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the 2021 Natural Science Basic Research Project of Shaanxi Province, No.: 2021JQ-765.

Data Availability Statement: The data used to support the findings of this study are available from the corresponding author upon request.

Acknowledgments: The authors also thank anonymous reviewers for fruitful discussion and useful comments.

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

References

1. Hansen, S.B.; Padfield, R.; Syayuti, K.; Evers, S.; Zakariah, Z.; Mastura, S. Trends in global palm oil sustainability research. *J. Clean. Prod.* **2015**, *100*, 140–149. [CrossRef]
2. Nambiappan, B.; Ismail, A.; Hashim, N.; Ismail, N.; Shahari, D.N.; Idris, N.A.N.; Kamalrudin, M.S.; Nur, A.M.H.; Kushairi, A. Malaysia: 100 years of resilient palm oil economic performance. *J. Oil Palm Res.* **2018**, *30*, 13–25. [CrossRef]
3. Maluin, F.N.; Hussein, M.Z.; Idris, A.S. An overview of the oil palm industry: Challenges and some emerging opportunities for nanotechnology development. *Agronomy* **2020**, *10*, 356. [CrossRef]
4. Cisternas, I.; Velásquez, I.; Caro, A.; Rodríguez, A. Systematic literature review of implementations of precision agriculture. *Comput. Electron. Agric.* **2020**, *176*, 105626. [CrossRef]
5. Miranda, V.; Pina, P.; Heleno, S.; Vieira, G.; Mora, C.; Schaefer, C.E. Monitoring recent changes of vegetation in Fildes Peninsula (King George Island, Antarctica) through satellite imagery guided by UAV surveys. *Sci. Total Environ.* **2020**, *704*, 135295. [CrossRef]
6. Choi, H.S.; Kim, E.M. Automatic geo-referencing of sequential drone images using linear features and distinct points. *J. Korean Soc. Surv. Geod. Photogramm. Cartogr.* **2019**, *37*, 19–28. [CrossRef]
7. Hamada, H.M.; Jokhio, G.A.; Al-Attar, A.A.; Yahaya, F.M.; Muthusamy, K.; Humada, A.M.; Gul, Y. The use of palm oil clinker as a sustainable construction material: A review. *Cem. Concr. Compos.* **2020**, *106*, 103447. [CrossRef]
8. Rehman, T.U.; Mahmud, M.S.; Chang, Y.K.; Jin, J.; Shin, J. Current and future applications of statistical machine learning algorithms for agricultural machine vision systems. *Comput. Electron. Agric.* **2019**, *156*, 585–605. [CrossRef]
9. 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]
10. Manandhar, A.; Hoegner, L.; Stilla, U. Palm tree detection using circular autocorrelation of polar shape matrix. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *3*, 465–472. [CrossRef]
11. Malek, S.; Bazi, Y.; Alajlan, N.; AlHichri, H.; Melgani, F. Efficient framework for palm tree detection in UAV images. *IEEE J. Selec. Top. Appl. Earth Obser. Remote Sens.* **2014**, *7*, 4692–4703. [CrossRef]
12. Shrestha, A.; Mahmood, A. Review of deep learning algorithms and architectures. *IEEE Access* **2019**, *7*, 53040–53065. [CrossRef]
13. 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]

14. Zortea, M.; Nery, M.; Ruga, B.; Carvalho, L.B.; Bastos, A.C. Oil-Palm Tree Detection in Aerial Images Combining Deep Learning Classifiers. In Proceedings of the IGARSS 2018—2018 IEEE International Geoscience and Remote Sensing Symposium, Valencia, Spain, 22–27 July 2018; pp. 657–660. Available online: https://ieeexplore.ieee.org/abstract/document/8519239 (accessed on 1 May 2020).
15. Mubin, N.A.; Nadarajoo, E.; Shafri, H.Z.M.; Hamedianfar, A. Young and mature oil palm tree detection and counting using convolutional neural network deep learning method. *Int. J. Remote Sens.* **2019**, *40*, 7500–7515. [CrossRef]
16. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 1137–1149. [CrossRef] [PubMed]
17. Benjdira, B.; Khursheed, T.; Koubaa, A.; Ammar, A.; Ouni, K. Car Detection Using Unmanned Aerial Vehicles: Comparison between Faster R-CNN and YOLOv3. In Proceedings of the 2019 1st International Conference on Unmanned Vehicle Systems-Oman (UVS), Muscat, Oman, 5–7 February 2019; pp. 1–6. Available online: https://ieeexplore.ieee.org/document/8658300 (accessed on 1 May 2020).
18. Neupane, B.; Horanont, T.; Hung, N.D. Deep learning based banana plant detection and counting using high-resolution red-green-blue (RGB) images collected from unmanned aerial vehicle (UAV). *PLoS ONE* **2019**, *14*, e0223906. [CrossRef]
19. Bai, T.; Pang, Y.; Wang, J.; Han, K.; Luo, J.; Wang, H.; Zhang, H. An optimized Faster R-CNN method based on DRNet and RoI align for building detection in remote sensing images. *Remote Sens.* **2020**, *12*, 762. [CrossRef]
20. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
21. Liu, X.; Ghazali, K.H.; Han, F.; Mohamed, I.I. Automatic detection of oil palm tree from UAV images based on the deep learning method. *Appl. Artif. Intell.* **2020**, *35*, 13–24. [CrossRef]
22. Lee, S.J.; Kim, S.H.; Kim, H.J. Robust translational force control of multi-rotor UAV for precise acceleration tracking. *IEEE Trans. Autom. Sci. Eng.* **2019**, *17*, 562–573. [CrossRef]
23. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2015**, arXiv:1409.1556.
24. Kounalakis, T.; Triantafyllidis, G.A.; Nalpantidis, L. Deep learning-based visual recognition of Rumex for robotic precision farming. *Comput. Electron. Agric.* **2019**, *165*, 104973. [CrossRef]
25. Sun, X.; Wu, P.; Hoi, S.C. Face detection using deep learning: An improved faster RCNN approach. *Neurocomputing* **2018**, *299*, 42–50. [CrossRef]
26. He, K.; Sun, J. Convolutional Neural Networks at Constrained Time Cost. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 5353–5360.
27. Nair, V.; Hinton, G.E. Rectified Linear Units Improve Restricted Boltzmann Machines. In Proceedings of the International Conference on Machine Learning (ICML), Haifa, Israel, 21–24 June 2010.
28. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
29. Ketkar, N. Stochastic Gradient Descent. In *Deep Learning with Python*; Apress: Berkeley, CA, USA, 2017; pp. 113–132.
30. Dalponte, M.; Ene, L.T.; Marconcini, M.; Gobakken, T.; Næsset, E. Semi-supervised SVM for individual tree crown species classification. *ISPRS J. Photogramm. Remote Sens.* **2017**, *110*, 77–87. [CrossRef]
31. Ke, Y.; Quackenbush, L.J. A review of methods for automatic individual tree-crown detection and delineation from passive remote sensing. *Int. J. Remote Sens.* **2011**, *32*, 4725–4747. [CrossRef]
32. Obuchowicz, R.; Oszust, M.; Bielecka, M.; Bielecki, A.; Piórkowski, A. Magnetic resonance image quality assessment by using non-maximum suppression and entropy analysis. *Entropy* **2020**, *22*, 220. [CrossRef]