remote sensing

MDPI

Article

Tree Species Classification in UAV Remote Sensing Images Based on Super-Resolution Reconstruction and Deep Learning

Yingkang Huang [1,2], Xiaorong Wen [1,2,*,†], Yuanyun Gao [3,†], Yanli Zhang [4] and Guozhong Lin [2]

[1] Co-Innovation Center for Sustainable Forestry in Southern China, Nanjing Forestry University, Nanjing 210037, China; kerwin@njfu.edu.cn
[2] College of Forestry, Nanjing Forestry University, Nanjing 210037, China
[3] Ministry of Ecology and Environment, Nanjing Institute of Science, Nanjing 210037, China; gaoyuanyun@nies.org
[4] Arthur Temple College of Forestry and Agriculture, Stephen F. Austin State University, Nacogdoches, TX 75962, USA
* Correspondence: wxr9872@njfu.edu.cn
† These authors contributed equally to this work.

Abstract: We studied the use of self-attention mechanism networks (SAN) and convolutional neural networks (CNNs) for forest tree species classification using unmanned aerial vehicle (UAV) remote sensing imagery in Dongtai Forest Farm, Jiangsu Province, China. We trained and validated representative CNN models, such as ResNet and ConvNeXt, as well as the SAN model, which incorporates Transformer models such as Swin Transformer and Vision Transformer (ViT). Our goal was to compare and evaluate the performance and accuracy of these networks when used in parallel. Due to various factors, such as noise, motion blur, and atmospheric scattering, the quality of low-altitude aerial images may be compromised, resulting in indistinct tree crown edges and deficient texture. To address these issues, we adopted Real-ESRGAN technology for image super-resolution reconstruction. Our results showed that the image dataset after reconstruction improved classification accuracy for both the CNN and Transformer models. The final classification accuracies, validated by ResNet, ConvNeXt, ViT, and Swin Transformer, were 96.71%, 98.70%, 97.88%, and 98.59%, respectively, with corresponding improvements of 1.39%, 1.53%, 0.47%, and 1.18%. Our study highlights the potential benefits of Transformer and CNN for forest tree species classification and the importance of addressing the image quality degradation issues in low-altitude aerial images.

Keywords: self-attention mechanism networks; convolutional neural networks; Real-ESRGAN; transformer; tree species classification

Citation: Huang, Y.; Wen, X.; Gao, Y.; Zhang, Y.; Lin, G. Tree Species Classification in UAV Remote Sensing Images Based on Super-Resolution Reconstruction and Deep Learning. *Remote Sens.* **2023**, *15*, 2942. https://doi.org/10.3390/rs15112942

Academic Editors: Moulay A. Akhloufi and Mozhdeh Shahbazi

Received: 4 May 2023
Revised: 31 May 2023
Accepted: 2 June 2023
Published: 5 June 2023

1. Introduction

Traditional tree species classification methods primarily rely on the expertise of forest workers who visually identify and judge trees based on features such as leaf shape, crown shape, and texture. These methods are often subjective and labor-intensive, requiring extensive fieldwork and manual identification. To improve the accuracy of tree species classification, LiDAR data are often combined with hyperspectral (HS) data [1–3]. However, the processing of airborne LiDAR data can be costly and complex, which makes this method unsuitable for large-scale forest classification [4]. Conventional research methods for tree species classification include manual feature extraction and classical machine learning algorithms, such as Support Vector Machines (SVMs) [5–7], Artificial Neural Networks (ANNs) [8,9], and Random Forest (RF) [10–12]. Burai, P. et al. [13] used airborne HS imagery and image classification methods (multi-label classification and SVM) combined with feature extraction to discriminate between species and clones of energy trees. They proposed an adaptive binary tree SVM classifier (ABTSVM) to improve the species-level

classification accuracy. Rocha, S.J.S.S.D. et al. [14] used ANNs based on competition index and climatic and categorical variables to predict tree survival and mortality in the semideciduous seasonal forests of the Atlantic Forest biome, reaching a high classification performance. Freeman, E.A. et al. [15] proposed strategies to address the issues of varying sampling intensity across different strata and the imbalanced presence of target species in training data when using the RF model for species distribution modeling. However, traditional machine learning methods for tree species classification often rely on handcrafted features, including various vegetation indices and texture features. In contrast, CNNs offer a significant advantage over traditional approaches in extracting essential features from raw data, which enables their widespread application in tree species classification tasks.

Since LeCun, Y. and his colleagues pioneered the use of CNNs for image classification tasks in 1998 [16], and it has been demonstrated that CNNs have significant advantages in extracting low-level features and visual structures. More recently, Krizhevsky, A. and his team introduced a deep CNN architecture called AlexNet [17], which was trained on a large scale using GPUs and obtained breakthrough results on the ImageNet dataset. At the same time, the development of low-altitude UAVs has made the acquisition of high-resolution aerial images with richer texture information than satellite images easier [18]. This combination has resulted in significant advances in tree species classification. For example, Nezami, S. et al. [19] proposed a deep learning method based on 3D-CNN for the high-accuracy classification of three major tree species in a boreal forest using RGB and HS data layers. Kapil, R. et al. [20] proposed a RetinaNet-based method that reached an average accuracy of 98.95% in classifying different stages of bark beetle attacks on individual trees. Hu, M. et al. [21] used a transfer-learning-based approach that fused multiple deep learning models to solve tree species classification in complex backgrounds, attaining an overall average accuracy of 93.75%. Natesan, S. et al. [22] used DenseNet for classifying forest tree species at the individual tree level using high-resolution RGB images from UAVs. The validation results demonstrate an accuracy of over 84% in distinguishing coniferous tree species in eastern Canada. Ford, D. J. [23] delved into the use of high-resolution RGB imagery from UAVs for tree species classification in a tropical wet forest. The study compared three classifiers and found that U-Net obtained the highest overall accuracy of 71.2%, suggesting the suitability of CNNs for fine-grained species-level classification using UAV data. In addition, some researchers have used deep learning in combination with UAV-Borne LiDAR data for individual tree crown segmentation studies [24]. However, it is noted that canopy images are different from natural images, and mutual relationships between tree canopies can affect classification results in forests with medium to high canopy density. In addition, low-altitude drone canopy images exhibit intra-canopy heterogeneity at the level of individual tree crowns, while displaying repeated similarity at the overall level. Additionally, due to hardware limitations of the drone imaging equipment, the images may suffer from blurred tree crown boundaries and weak texture factors. Therefore, simply adopting a residual network model to identify tree species can lead to overlooking the importance of different channel feature maps for classification results and result in a limitation of accuracy. However, attention mechanism models have a natural long-range modeling ability that enables the full utilization of effective global information from shallow to deep layers. By taking advantage of this ability, researchers can more effectively classify tree species and overcome the unique challenges posed by canopy images.

The goal of this research article is to investigate tree species classification methods using both CNN models and Transformer models on UAV tree crown images. Our proposed approach involves several steps. Firstly, we utilized a sliding window cropping technique on low-altitude drone canopy images to obtain smaller image patches. Then, we applied Real-ESRGAN technology for super-resolution reconstruction to restore the blurry canopy images and enhance their spatial resolution. Next, we made use of both CNN models and Transformer models to extract features from the tree crown images, and performed a differential comparative analysis to evaluate the performance of these two approaches. Due to the relatively modest scale of the canopy sample set in this experiment, and taking

into account the constraints imposed by computational resources, this study opted to train and validate the ResNet-50, ConvNeXt-T, Swin-T, and ViT-B models. These models boast fewer parameters and exhibit lower computational intricacy, thereby enhancing the efficacy of the training and inference processes. Through this efficient and accurate tree species classification method based on low-altitude aerial tree crown images, we aim to delve into the potential applications of artificial intelligence in forestry intelligence and information construction, and explore the possibilities of improving tree species classification accuracy using advanced deep learning techniques.

2. Study Area and Data Collection

2.1. Study Area

The study area was located in Dongtai Forest Farm, Yancheng City, Jiangsu Province, with geographical coordinates ranging from 120°47′11″E to 120°52′0″E and 32°53′30″N to 32°51′17″N, as shown in Figure 1. We used a Liortho high-resolution imaging system mounted on a Digital Green Earth octocopter UAV for data collection, with a flight altitude of 200 m and image resolution of 0.2 m. In the Dongtai Forest Farm study area, we picked out three representative plots as experimental areas. From these areas, we selected four predominant tree species, Poplar, Metasequoia, Bamboo, and Ginkgo, as the primary subjects of our research.

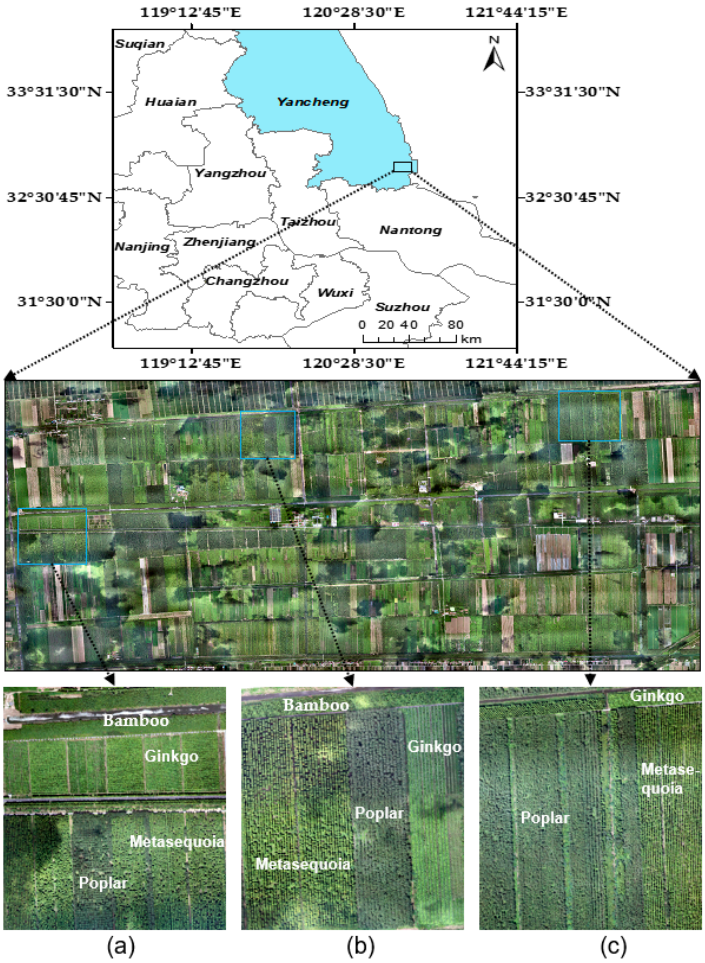

Figure 1. Study area: Dongtai Forest Farm. (**a**–**c**) UAV-RGB images of the main tree species for the three experimental areas, respectively.

2.2. Data Collection and Preprocessing

In this study, we selected four tree species, namely Metasequoia, Bamboo, Poplar, and Ginkgo, as the target recognition objects within the study area. To collect samples, we used sliding window cropping with a window size of 64 × 64 pixels, resulting in a total of 4338 samples. Specifically, the number of samples obtained for each tree species was as follows: 1101 for Bamboo, 880 for Ginkgo, 1105 for Metasequoia, and 1252 for Poplar. These samples were affected by various imaging factors, such as forest canopy closure, tree species, lighting, background, and shooting height, resulting in differences in intra-class and inter-class characteristics, as depicted in Figure 2.

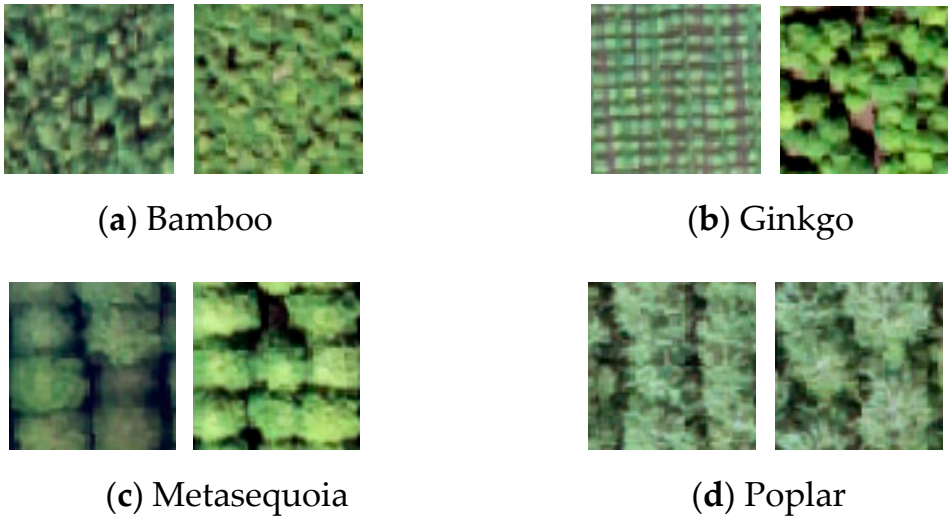

Figure 2. Sample tree species: (**a**) Bamboo; (**b**) Ginkgo; (**c**) Metasequoia; (**d**) Poplar.

Given that neural network models, especially Transformer models and their variants, can have millions or even billions of parameters, it is crucial to have sufficient samples for effective training. To increase the sample size, we performed data augmentation on the training and validation sample sets. The data augmentation techniques applied to the tree canopy image samples included random rotations by 90°, 180°, and 270°, horizontal flipping, and brightness adjustment as depicted in Figure 3. These transformations were implemented using the 'RandomRotation', 'RandomHorizontalFlip', and 'ColorJitter' functions provided by the Torchvision library. By incorporating these techniques into the data augmentation pipeline, variations were introduced to the dataset, allowing for enhanced training and improved model generalization. Random rotations provided diverse perspectives of the tree canopies, while horizontal flipping increased the dataset's diversity by mirroring the images. Additionally, brightness adjustment ensured robustness to varying lighting conditions, enabling the model to generalize better across different environments. These techniques collectively contributed to the overall robustness and performance of the tree canopy classification model.

Furthermore, we randomly divided the samples into training and validation sets in an 8:2 ratio, respectively. This ensured that both sets had a representative distribution of samples from each tree species, allowing for robust model evaluation and performance estimation. The augmented dataset with increased sample size and diversity, along with the appropriate training and validation set partitioning, facilitated the training of the neural network models for accurate tree species recognition in the study area.

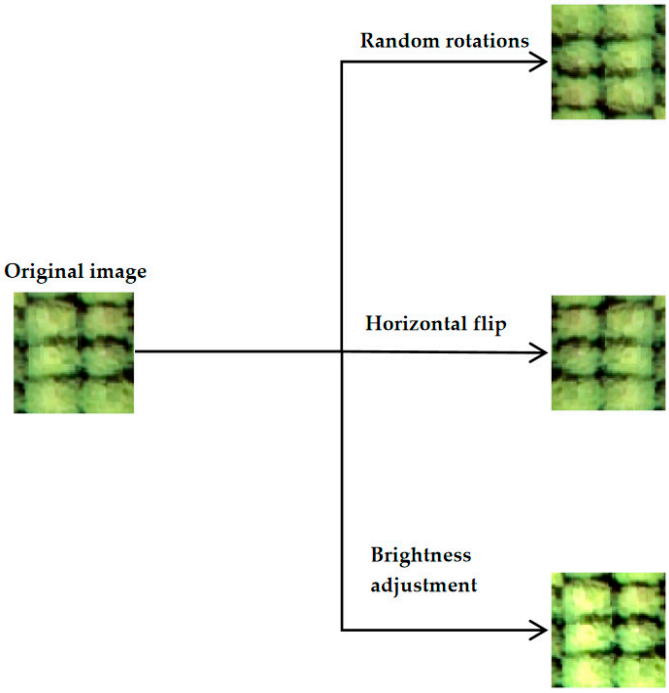

Figure 3. Schematic diagram of canopy sample data augmentation.

3. Methodology

3.1. Super-Resolution Reconstruction

Faced with the limitations of hardware equipment, such as high-denseness forest stands, aerial photography angles, and resolution, as well as complex geographic backgrounds, drone aerial images often suffer from degradation issues such as lost details, reduced brightness, and partially blurred tree crowns. To solve these problems, this paper uses Real-ESRGAN technology [25] to perform super-resolution reconstruction, denoising, and deblurring on the original tree crown images. Classical first-order degradation models, as expressed in Equation (1), often consider only one type of degradation operation, such as blur or noise, while ignoring the simultaneous presence of multiple degradation operations. However, in practical scenarios, images may undergo multiple degradations simultaneously. For example, during transmission, images may experience blur, resolution reduction, and the introduction of noise. In order to more accurately simulate the degradation process of real images, Real-ESRGAN proposes a high-order degradation model, as expressed in Equation (2), which utilizes multiple repeated degradation processes, with each degradation process representing a classical degradation model. Through ablation experiments, the second-order model has been proven to exhibit excellent performance and practicality, effectively meeting the requirements of most image processing tasks [25]. Real-ESRGAN's second-order degradation model, as shown in Figure 4, aims to simulate the degradation process of real images through several steps, including blur processing, downsampling, noise addition, and JPEG compression. Firstly, in the blur processing stage, isotropic and anisotropic Gaussian blur kernels are applied to the original image, resulting in the loss of details and clarity to mimic real-world blurring effects. Subsequently, the blurred image undergoes downsampling using randomly selected methods such as bilateral interpolation, bilinear interpolation, and regional interpolation, further reducing the image's resolution and diminishing details and clarity to simulate actual resolution reduction effects. Next, noise is added based on the image type (color or grayscale). For color images, both Gaussian noise and noise following the Poisson distribution are added to simulate environmental and sensor noise, while grayscale images only receive Gaussian noise. Finally, the image is subjected to JPEG compression based on a compression quality

parameter ranging from 0 to 100, where lower compression quality leads to poorer image quality and more severe distortion.

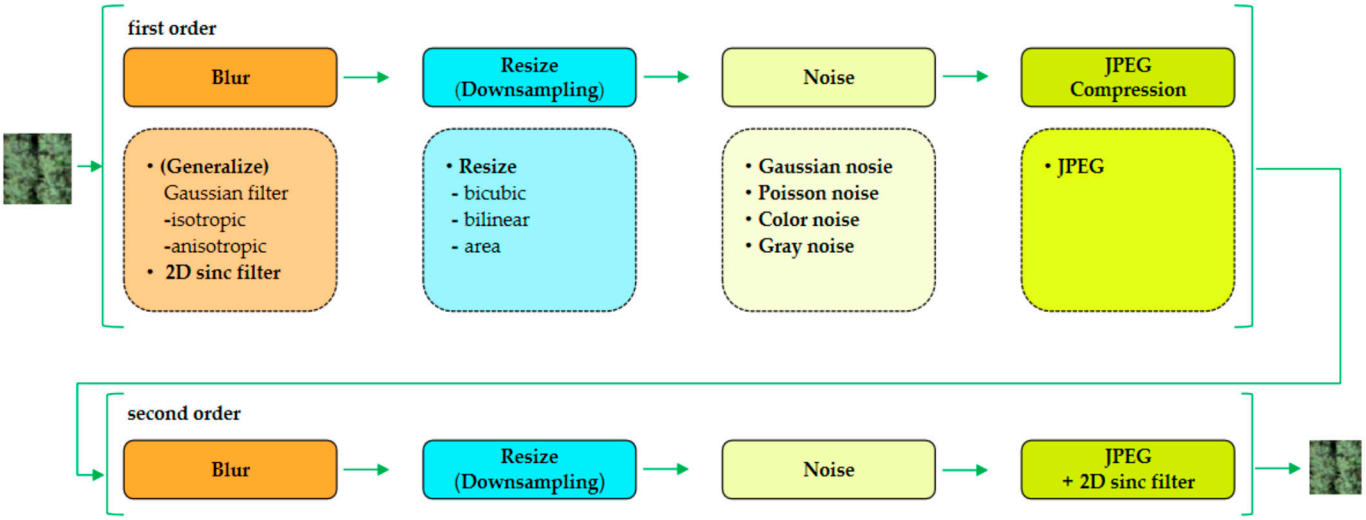

Figure 4. Real-ESRGAN second-order degradation model.

Both in the initial blur processing step and the final synthesis step, a sinc filter is used. The expression for the sinc filter is shown in Equation (3). The sinc filter exhibits high selectivity in the frequency domain, responding differently to signals of various frequencies. Consequently, the sinc filter can smooth and blur specific frequency details in the image, thereby reducing its details and clarity. Additionally, the sinc filter possesses inverse filtering properties, allowing it to repair blurred or degraded images during the restoration process, mitigating degradation effects and improving image quality. It is worth noting that in the final synthesis step, the order of applying the sinc filter and JPEG compression is randomly exchanged to cover a broader range of degradation scenarios.

In summary, Real-ESRGAN's second-order degradation model adopts various degradation operations, such as blur processing, downsampling, noise addition, and JPEG compression, to simulate the degradation process of real images. These processed degraded images lose their detail and clarity, exhibiting visual effects such as blurring, reduced resolution, noise addition, and distortion, thereby providing challenging inputs for the subsequent Real-ESRGAN super-resolution reconstruction process.

$$x = D(y) = [(y \otimes k) \downarrow_r + n]_{JPEG} \tag{1}$$

$D(\cdot)$ denotes the degradation process, y denotes the input image, k denotes the blur function, $\downarrow_r$ denotes the downsampling factor, n denotes the noise, and $[\,]_{JPEG}$ denotes the compression of the obtained result in JPEG format.

$$x = D^n(y) = (D_n \circ \cdots \circ D_2 \circ D_1)(y) \tag{2}$$

The above equation is actually a multiple repetition operation of the first-order degradation, where each D represents the execution of one first-order degradation.

$$k(i,j) = \frac{\omega_c}{2\pi\sqrt{i^2 + j^2}} J_1(\omega_c \sqrt{i^2 + j^2}) \tag{3}$$

(i, j) is the kernel coordinate, ω_c is the cutoff frequency, and J_1 is a first-order Bessel function of the first type.

Real-ESRGAN builds upon the generator structure of ESRGAN [26] as a foundation and further enhances and refines it through design improvements, comprising numerous Residual-in-Residual Dense Blocks (RRDB) for enhanced performance. These processed

images, obtained through the preceding pre-processing steps elucidated earlier, are subsequently channeled into the generator of ESRGAN, as demonstrated in Figure 5 underneath. Primarily, pixel shuffling is implemented to diminish the spatial dimensions and augment the channel properties. Then, the resultant outcome is fed into the principal architecture of ESRGAN for super-resolution reconstruction.

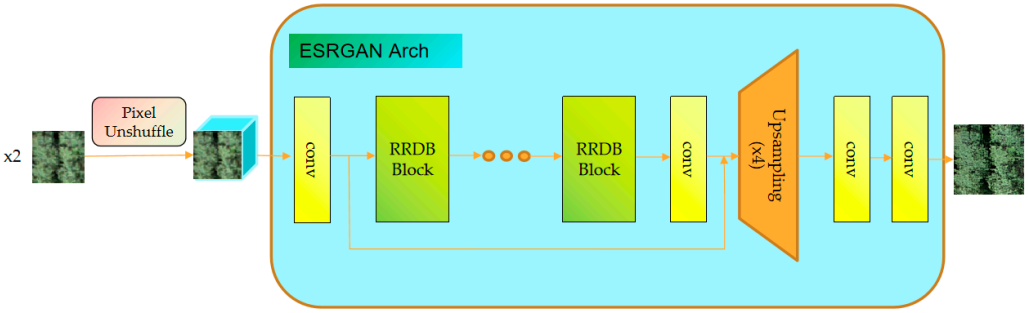

Figure 5. ESRGAN architecture. We used a pixel-unshuffle operation to diminish the spatial dimensions and re-arrange information to the channel dimension for scale factors of x2.

Moreover, owing to Real-ESRGAN's aspiration to confront a considerably wider range of degradations than ESRGAN, the original VGG-style discriminator design in ESRGAN is no longer suitable. Instead, Real-ESRGAN introduces a U-Net framework with skip connections for the discriminator, inspired by the referenced research endeavors [27,28]. Finally, the generated images are mixed with the input images and fed into the discriminator for discrimination, which uses spectrally normalized U-Net to mitigate excessive sharpness and artifacts introduced by GAN training. The original tree species canopy images were restored by Real-ESRGAN super-resolution reconstruction as shown in Figure 6 below, and the canopy texture details were restored and processed to facilitate the neural network model to extract clear canopy edge contour features and detailed texture features.

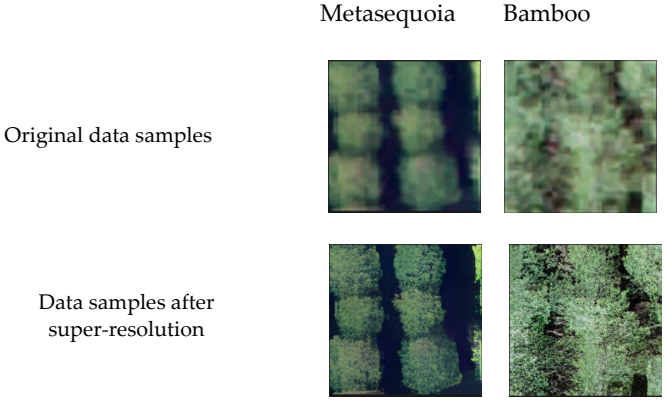

Figure 6. Comparison of original canopy images and super-resolution reconstructed canopy images.

3.2. Model Training

3.2.1. ResNet Model

Residual blocks proposed by ResNet [29], as shown in Figure 7, can effectively solve the problem of network degradation in deep networks. Two varieties of shortcut connections are adopted within the layers of ResNet. Identity shortcuts are utilized when the dimensions of the input and output are equal, while projection shortcuts are used to align dimensions [29]. In this paper, we use ResNet-50 as a representative of such models. The decision to adopt ResNet-50 was grounded on empirical observations and experiments conducted by the researchers who proposed the ResNet architecture. Their findings revealed that surpassing a certain depth threshold in the network resulted in diminishing performance improvements or even a decline

in accuracy, primarily due to the issue of vanishing gradients [29]. ResNet-50 successfully strikes an excellent balance between model depth and performance. As the name suggests, ResNet-50 consists of 50 layers, as shown in Figure 8, divided into five stages. Stage 0 can be regarded as the preprocessing of input data, while stages 1 to 4 are each composed of 3, 4, 6, and 3 bottleneck blocks, respectively. (1) In Stage 0, the first step was to convert the canopy image, which has a size of 224 × 224, into a digital matrix with dimensions [224, 224, 3]. Subsequently, we used a convolutional layer with a 7 × 7 kernel, stride of 2, and 64 output channels, resulting in a feature map measuring 112 × 112 × 64. Furthermore, a 3 × 3 max pooling layer with a stride of 2 was applied to reduce the feature map's size, yielding a 56 × 56 × 64 representation. (2) Moving forward, the 56 × 56 × 64 feature map underwent processing in Stage 1. At this stage, we adopted three bottleneck blocks to facilitate the integration process. Each block consisted of a sequence of convolutional operations, including 1 × 1 convolutions with 64 output channels, 3 × 3 convolutions with 64 output channels, and another 1 × 1 convolution with 256 output channels. These operations reshaped the feature map, resulting in a 56 × 56 × 256 representation. (3) Continuing the progression, the transformed 56 × 56 × 256 feature map proceeded to Stage 2, housing four bottleneck blocks. Each block implemented a series of convolutions to reshape the feature map from 56 × 56 × 256 to 28 × 28 × 512. (4) Continuing the sequence, the 28 × 28 × 512 feature map was inputted into Stage 3, containing six bottleneck blocks. Through a cascade of convolutions, the feature map underwent size transformation from 28 × 28 × 512 to 14 × 14 × 1024. (5) Subsequently, the modified 14 × 14 × 1024 feature map advanced to Stage 4, incorporating three bottleneck blocks. These convolutions modified the feature map's dimensions from 14 × 14 × 1024 to 7 × 7 × 2048. (6) Next, a global average pooling operation was applied to the 7 × 7 × 2048 feature map, computing the average value of each channel. Consequently, a feature map of size 1 × 1 × 2048 was obtained. (7) Finally, the 1 × 1 × 2048 feature map was flattened into a one-dimensional vector and subjected to processing through a fully connected layer for classification purposes. Given that this experiment encompasses four tree species, the output provides probability values for the four categories. The detailed architecture specifications of ResNet-50 are described in Table 1.

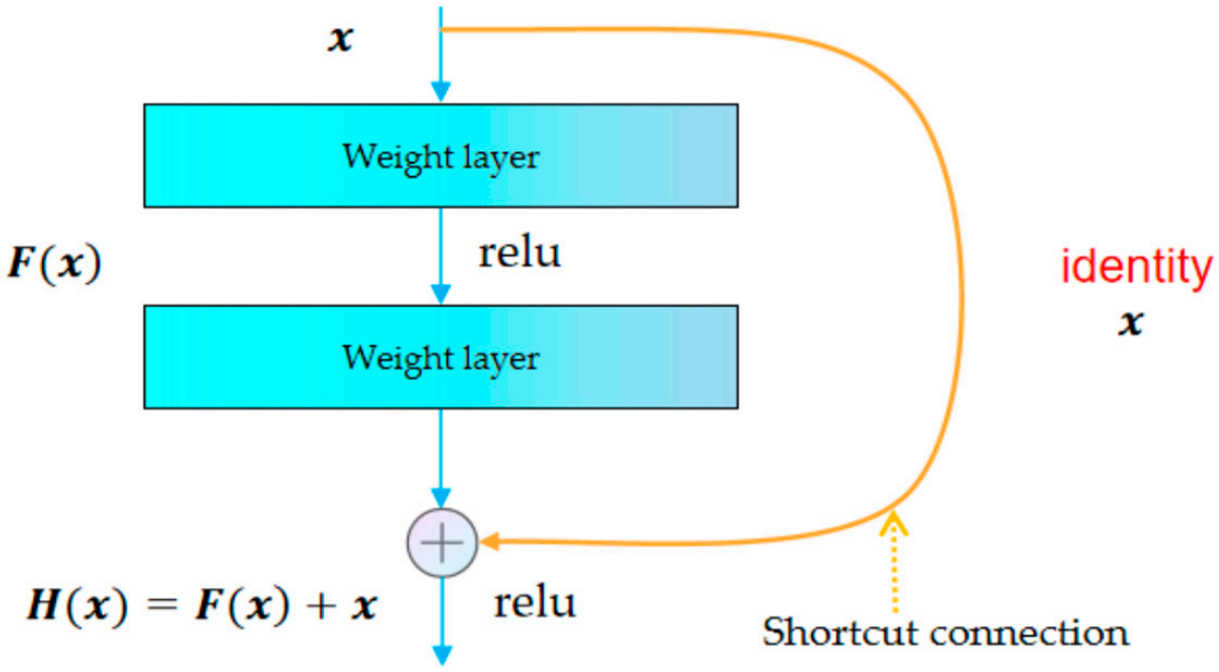

Figure 7. Residual block structure.

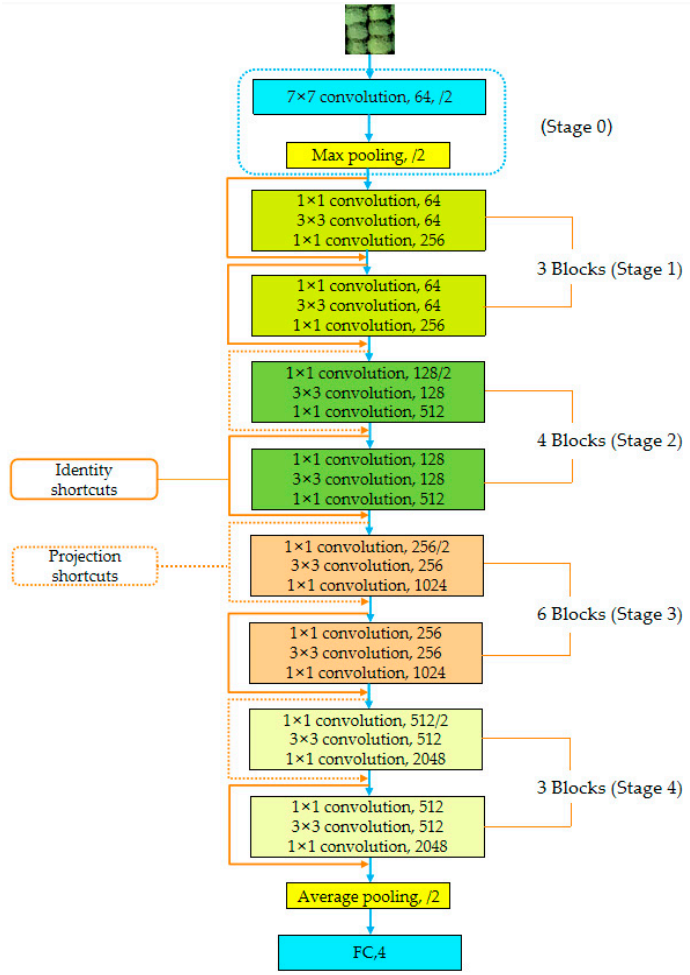

Figure 8. A depiction of the ResNet-50 network architecture. '/2' means stride is set to 2.

Table 1. Detailed architecture specifications for ResNet-50. '4-d fc' denotes a fully connected layer with 4 dimensions.

Stage	Output_Size	ResNet-50
Stage0	112 × 112	7 × 7, 64, stride 2
Stage1	56 × 56	3 × 3 max pooling, stride 2
		$\begin{bmatrix} 1\times1,\ 64 \\ 3\times3,\ 64 \\ 1\times1,\ 256 \end{bmatrix} \times 3$
Stage2	28 × 28	$\begin{bmatrix} 1\times1,\ 128 \\ 3\times3,\ 128 \\ 1\times1,\ 512 \end{bmatrix} \times 4$
Stage3	14 × 14	$\begin{bmatrix} 1\times1,\ 256 \\ 3\times3,\ 256 \\ 1\times1,\ 1024 \end{bmatrix} \times 6$
Stage4	7 × 7	$\begin{bmatrix} 1\times1,\ 512 \\ 3\times3,\ 512 \\ 1\times1,\ 2028 \end{bmatrix} \times 3$
	1 × 1	average pooling, 4-d fc, softmax

3.2.2. ConvNeXt Model

The ConvNeXt model [30] improves upon the ResNet architecture by incorporating ideas from the Transformer network [31], as depicted in Figure 9. Specifically, ConvNeXt implements techniques such as mimicking depthwise convolutions to form convolution blocks and referencing the design of the inverted bottleneck structure, resulting in enhanced performance. In this study, we made use of the ConvNeXt-T model. The process of classifying tree species with ConvNeXt-T involved the following steps: (1) To begin, we started with an input canopy image of size 224 × 224 × 3. This image was passed through a convolutional layer with a kernel size of 4 and a stride of 4. Afterward, a Layer Norm (LN) was applied to normalize the feature map, resulting in a 56 × 56 × 96-sized feature map. The LN operation plays a crucial role in enhancing network stability and generalization by standardizing the feature map. (2) Moving on, the feature map underwent four stages, each containing a ConvNeXt block. In each stage, there were 3, 3, 9, and 3 ConvNeXt blocks, respectively. These ConvNeXt blocks were composed of a 7 × 7 depthwise convolution with a stride of 1 and a padding of 3, followed by an LN. In addition, two 1 × 1 Conv2d layers with a stride of 1 and a GeLU activation function [32] were adopted. The output channels of the ConvNeXt blocks were 96, 192, 384, and 768, respectively. During the second to fourth stages, a downsampling operation was performed on the feature map. This downsampling operation included applying an LN and using a Conv2d layer with a stride of 2. This operation reduces the size of the feature map to half that of the previous layer. As a result, the feature map ended up containing 768 channels after passing through all four stages. (3) Next, the feature map was passed through a global average pooling layer, and a feature vector with a size of 1 × 1 × 768 was obtained. (4) The final step involved passing the feature vector through a fully connected layer to convert the 768-dimensional vector into an output vector of size 4. This output vector represents the probability values of the four tree species categories. Table 2 shows the detailed structural specifications of the ConvNeXt-T.

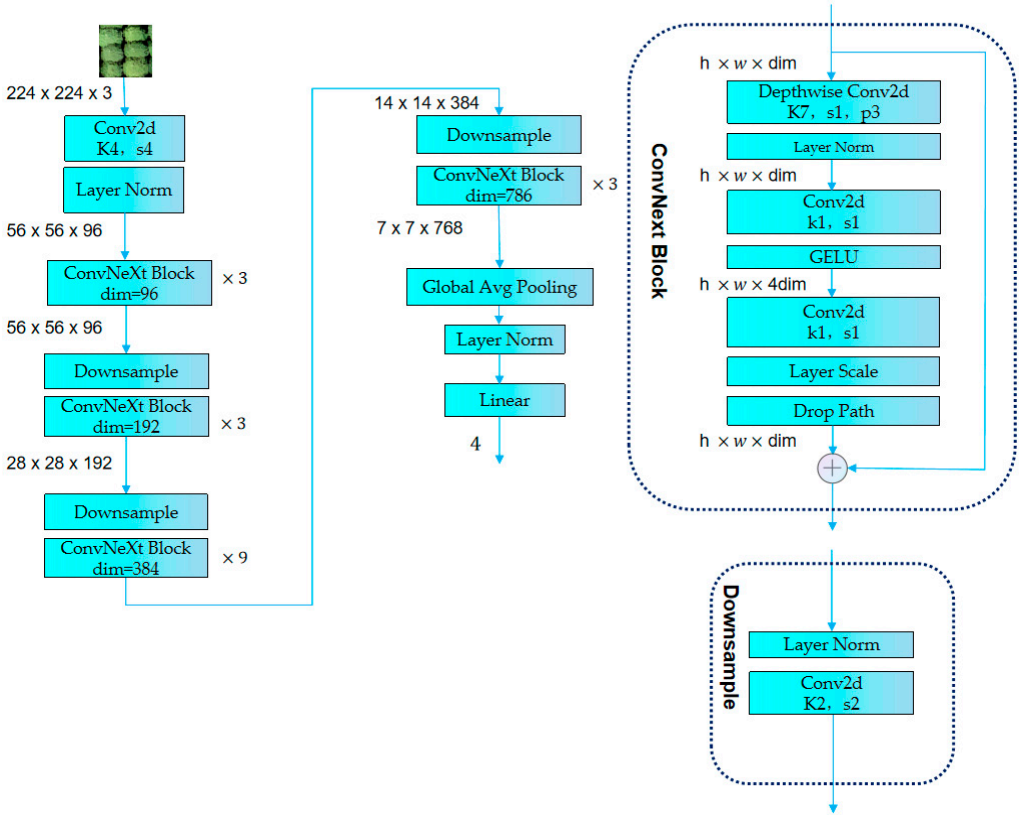

Figure 9. ConvNeXt-T Network Architecture.

Table 2. Detailed architecture specifications for ConvNeXt-T. 'd' is short for depthwise convolution.

Layer_Name	Output_Size	ConvNeXt-T
Conv1	56 × 56	4 × 4, 96, stride 4
Conv2_x	56 × 56	$\begin{bmatrix} \text{d}7 \times 7, 96 \\ 1 \times 1, 384 \\ 1 \times 1, 96 \end{bmatrix} \times 3$
Conv3_x	28 × 28	$\begin{bmatrix} \text{d}7 \times 7, 192 \\ 1 \times 1, 768 \\ 1 \times 1, 192 \end{bmatrix} \times 3$
Conv4_x	14 × 14	$\begin{bmatrix} \text{d}7 \times 7, 384 \\ 1 \times 1, 1536 \\ 1 \times 1, 384 \end{bmatrix} \times 9$
Conv5_x	7 × 7	$\begin{bmatrix} \text{d}7 \times 7, 768 \\ 1 \times 1, 3072 \\ 1 \times 1, 768 \end{bmatrix} \times 3$
	1 × 1	global average pooling, 4-d fc

3.2.3. ViT Model

The ViT model [33] is a specially designed Transformer model for image classification. It is composed of three modules: Embedding, Transformer Encoder, and Multi-Layer Perceptron (MLP) Head. The Transformer Encoder comprises LN, Multi-head Attention, Dropout, and MLP Block, and for this paper, we used the ViT-B model, as shown in Figure 10. Here is how the image classification process worked using ViT-B: (1) Initially, we input a tree canopy image with dimensions of 224 × 224 × 3. It underwent convolution using a 16 × 16 kernel and a stride of 16, resulting in a feature map sized 14 × 14 × 768. (2) Next, we flattened the feature map in both the height and width directions, transforming its size to 196 × 768. Subsequently, we concatenated a class token and applied positional encoding to the feature map, yielding a transformed size of 197 × 768. (3) Following this step, we applied Dropout and passed the input through 12 stacked Encoder Blocks. The output from the Encoder was then processed with LN, maintaining the feature map size at 197 × 768. We proceeded to extract the output corresponding to the class token and slice it, obtaining a vector of size 1 × 768. This vector was then fed into the MLP Head. (4) Finally, the feature vectors were fed into a fully connected layer with four neurons for classification, where each neuron represented a tree species category, resulting in the final classification results.

3.2.4. Swin Transformer Model

Due to the high computational cost and memory consumption of the self-attention mechanism in ViT models when processing high-resolution image tasks, the Swin Transformer model was proposed [34], which adopts a hierarchical structure as shown in Figure 11. The model comprises four key components: Patch Partition, Linear Embedding, Swin Transformer Block, and Patch Merging. Two successive Swin Transformer Blocks illustrated in Figure 12, incorporate Window-based Multi-head Self-attention (W-MSA) and Shifted Window-based Multi-head Self-attention (SW-MSA) to address memory consumption challenges while maintaining efficient performance. Additionally, Patch Merging adopts pooling-like operations to progressively reduce the feature map size and merge image blocks, constructing a hierarchical feature map in deeper layers. LN layers are used before each MSA module and each MLP, and residual connections are used after each MSA and MLP. These characteristics make it a versatile backbone for image classification and dense recognition tasks. In this study, we adopted the Swin-T model as a representative

example of such models. (1) Initially, the input was a tree crown image with dimensions 224 × 224 × 3, which underwent Patch Partition. This process involved dividing the image into fixed-sized blocks using a 4 × 4 convolutional kernel. The resulting feature map dimensions were 56 × 56 × 48. (2) Next, the Linear Embedding layer was subsequently applied to each channel of the pixel data, resulting in a linear transformation that changed the feature map dimensions to 56 × 56 × 96. (3) Moving forward, we proceeded to a series of Swin Transformer blocks. These blocks consisted of two variations: one utilizing the W-MSA structure and the other adopting the SW-MSA structure. Consequently, the Swin Transformer blocks appeared in even numbers, with 2 blocks in each of the first, second, and fourth stages, and 6 blocks in the third stage. The Swin Transformer block incorporated window partitioning and window reverse operations, maintaining the output feature map size at 56 × 56 × 96. Therefore, the input and output sizes of the Swin Transformer block remained unchanged. (4) Subsequently, Patch Merging was performed to reduce the spatial dimensions by half and double the channel count. This process was repeated across the four stages, eventually transforming the feature map dimensions to 7 × 7 × 768. (5) Finally, global average pooling was utilized to reduce the spatial dimension to 1, resulting in a feature vector of size 1 × 1 × 768. A linear classifier with four neurons was used to map the output vector to probability values corresponding to the four tree species categories, yielding the final prediction. The detailed Swin-T architecture specification is described in Table 3.

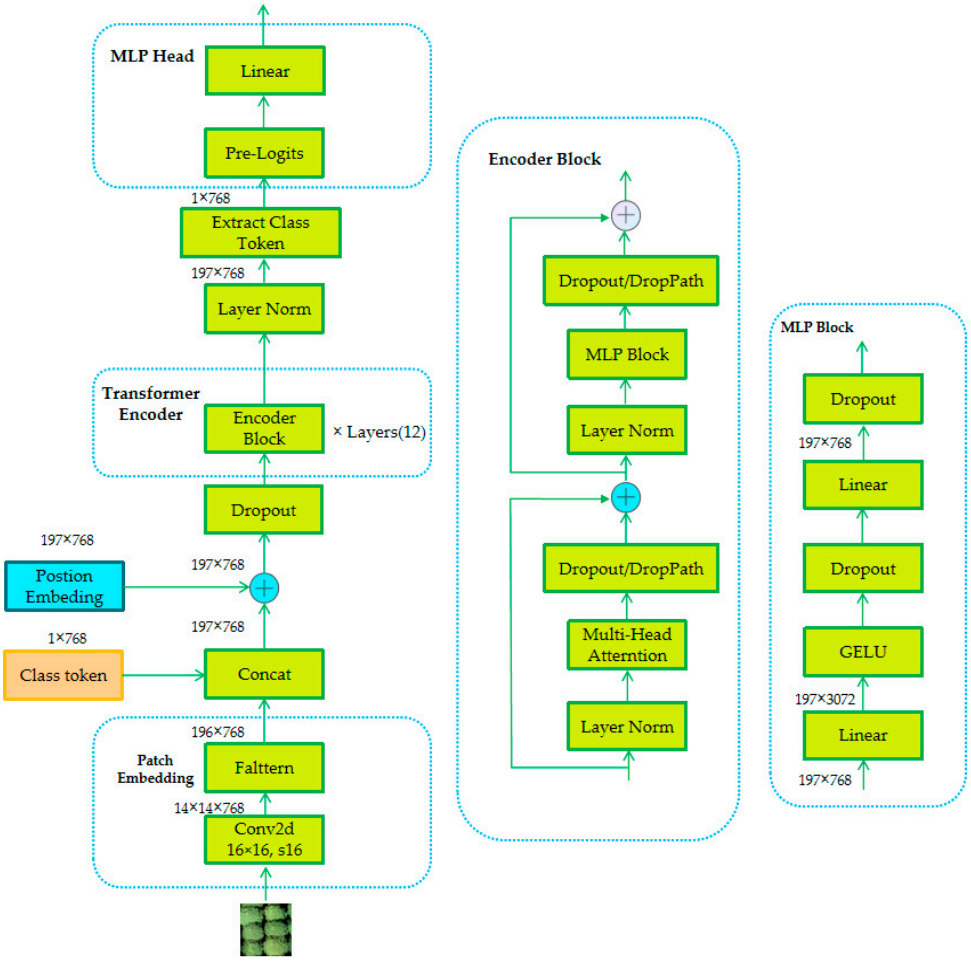

Figure 10. ViT-B Network Architecture.

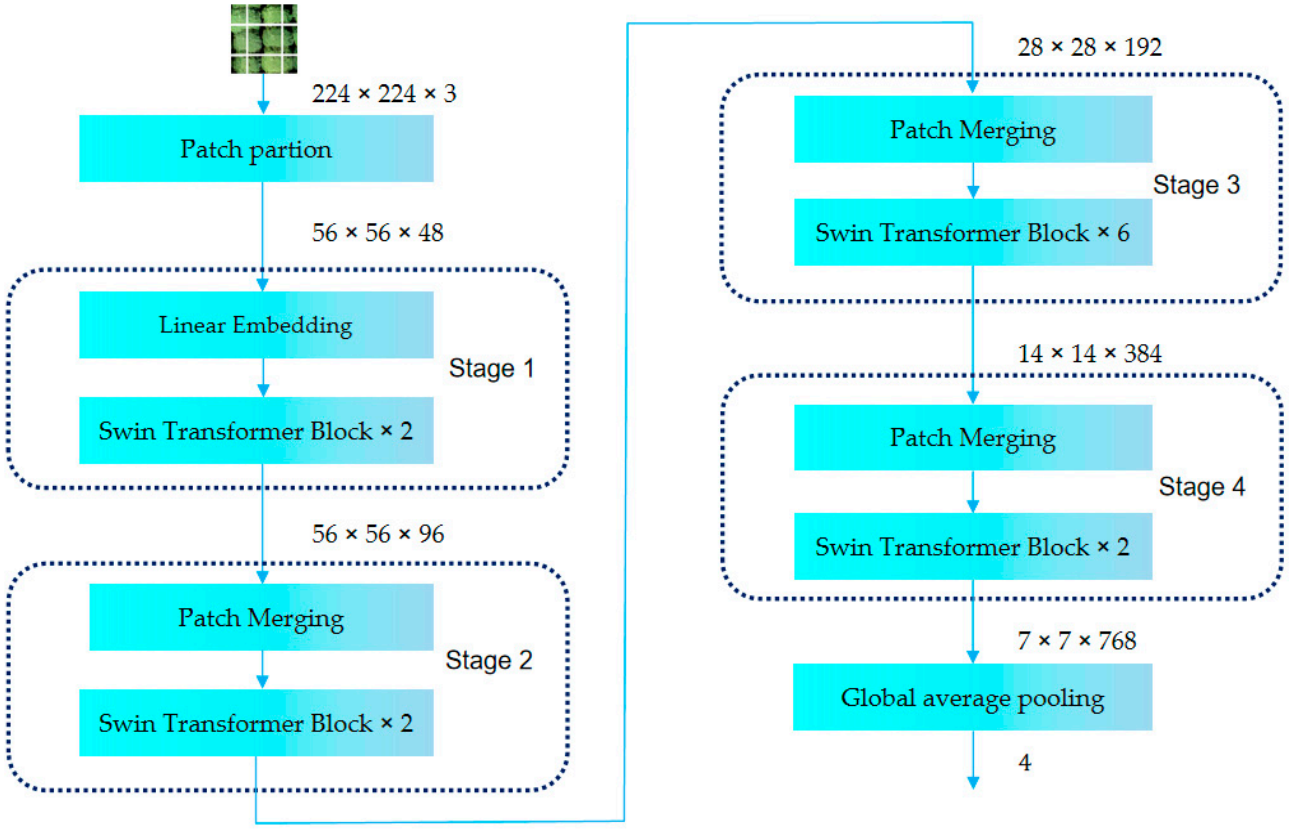

Figure 11. Swin-T Network Architecture.

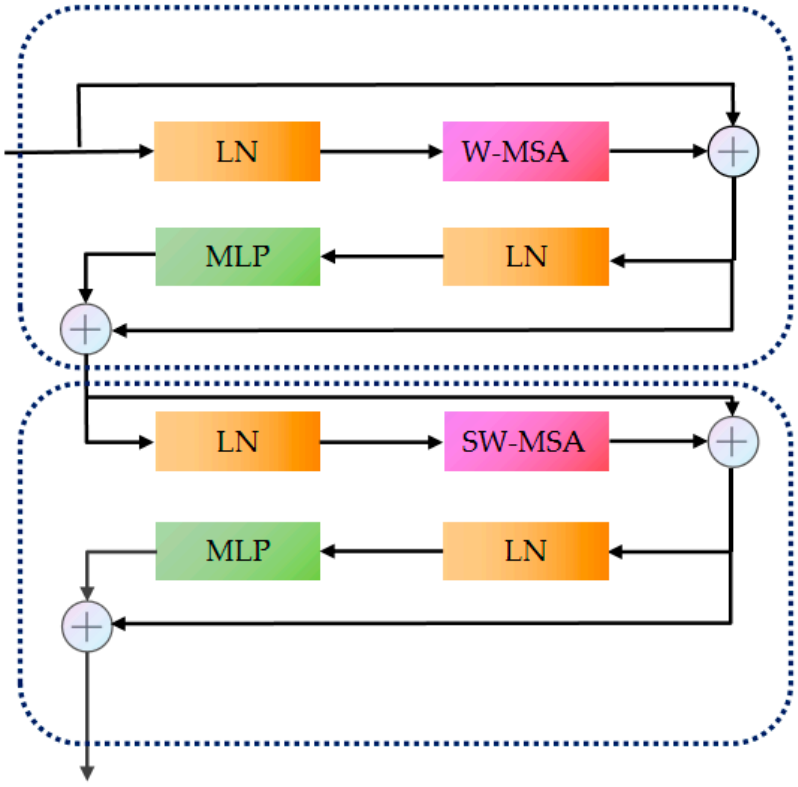

Figure 12. Two Successive Swin Transformer Blocks.

Table 3. Detailed architecture specifications for Swin-T. win. sz. indicates the size of the window used; dim indicates the channel depth of the feature map; head indicates the number of heads in a multi-headed attention module.

Stage_Name	Output_Size	Swin-T
Stage1	56 × 56	concat 4 × 4, 96, LN
		$\begin{bmatrix} \text{win. sz. } 7 \times 7 \\ \text{dim } 96, \text{head } 3 \end{bmatrix} \times 2$
Stage2	28 × 28	concat 2 × 2, 192, LN
		$\begin{bmatrix} \text{win. sz. } 7 \times 7 \\ \text{dim } 192, \text{head } 6 \end{bmatrix} \times 2$
Stage3	14 × 14	concat 2 × 2, 384, LN
		$\begin{bmatrix} \text{win. sz. } 7 \times 7 \\ \text{dim } 384, \text{head } 12 \end{bmatrix} \times 6$
Stage4	7 × 7	concat 2 × 2, 768, LN
		$\begin{bmatrix} \text{win. sz. } 7 \times 7 \\ \text{dim } 768, \text{head } 24 \end{bmatrix} \times 2$
	1 × 1	global average pooling, 4-d fc

3.3. Experimental Environment

The experiments were conducted on a computer with the following specifications: Windows 10 operating system, AMD Ryzen 5 5600X 6-Core Processor CPU, and NVIDIA GeForce RTX 3070Ti GPU with 8GB of memory. The deep learning platform used for training and evaluation was PyTorch 1.12.0, along with cudatoolkit11.3 for GPU acceleration. For raster and vector data processing, Arcgis10.8 was utilized, and Matplotlib 3.5.2 was used for data visualization. Python 3.9.13 was the programming language used for implementation and analysis.

4. Results

4.1. Comparison of CNN and Transformer Models for Tree Species Classification

The main objective of this investigation was to evaluate the impact of different models on tree species classification using canopy images. To this end, we selected two representative pre-trained models from both the CNN and Transformer models, specifically ResNet-50, ConvNeXt-T, ViT-B, and Swin-T. For these four models, we set the image input size to 224 × 224 pixels. During training, we used a batch size of 16 and the AdamW optimization algorithm, and trained for a total of 150 epochs. The initial learning rate was set to 4×10^{-4}, and the weight decay factor was set to 5×10^{-2}. We also used the LambdaLR strategy for learning rate adjustment.

To assess the performance of the models, we monitored the changes in the loss function and recognition accuracy of the original image data samples on the training set and validation set across 150 epochs. These changes are visualized in Figure 13 to understand the training progress and stability of each model. It is observed that for all models, the accuracy and loss tend to stabilize after around 120 to 130 epochs, with recognition classification accuracy exceeding 95%, indicating strong stability and high accuracy in the tree species classification task.

Furthermore, we evaluated the overall classification accuracy (OA), Kappa coefficient, and confusion matrix of each model, as shown in Figure 14. These evaluation metrics provide quantitative measures of the performance of the models in the tree species classification task. By analyzing these results, we can gain a comprehensive understanding of the performance of each model in tree species classification using canopy images.

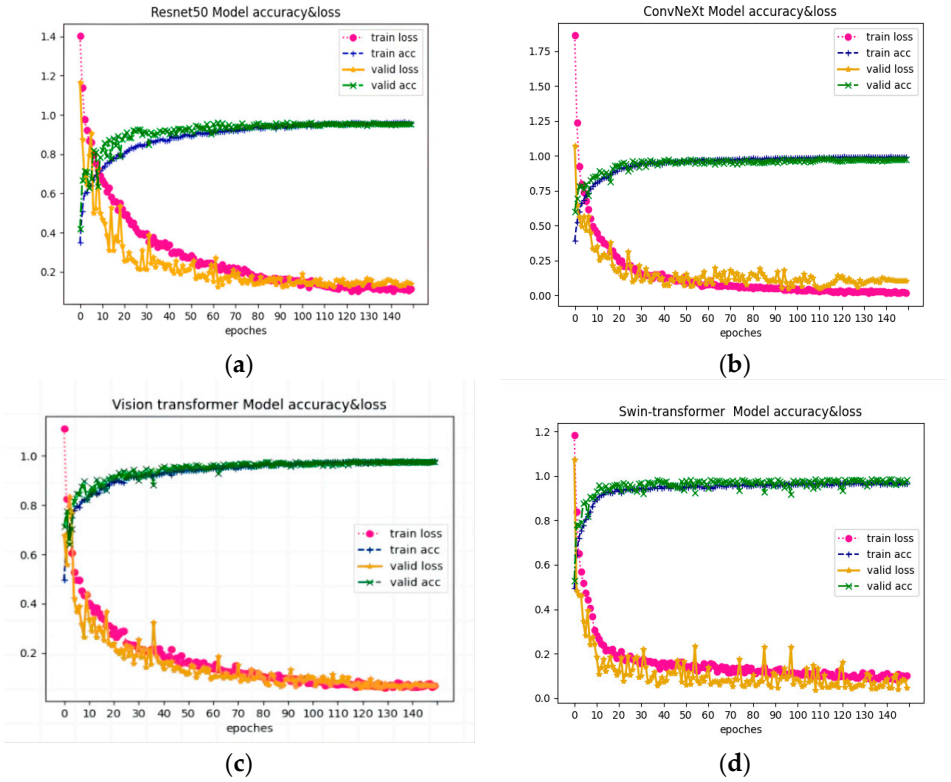

Figure 13. (**a**–**d**) are plots of the accuracy and loss rates of the ResNet-50, ConvNeXt-T, ViT-B, and Swin-T models trained and validated on the original dataset, respectively.

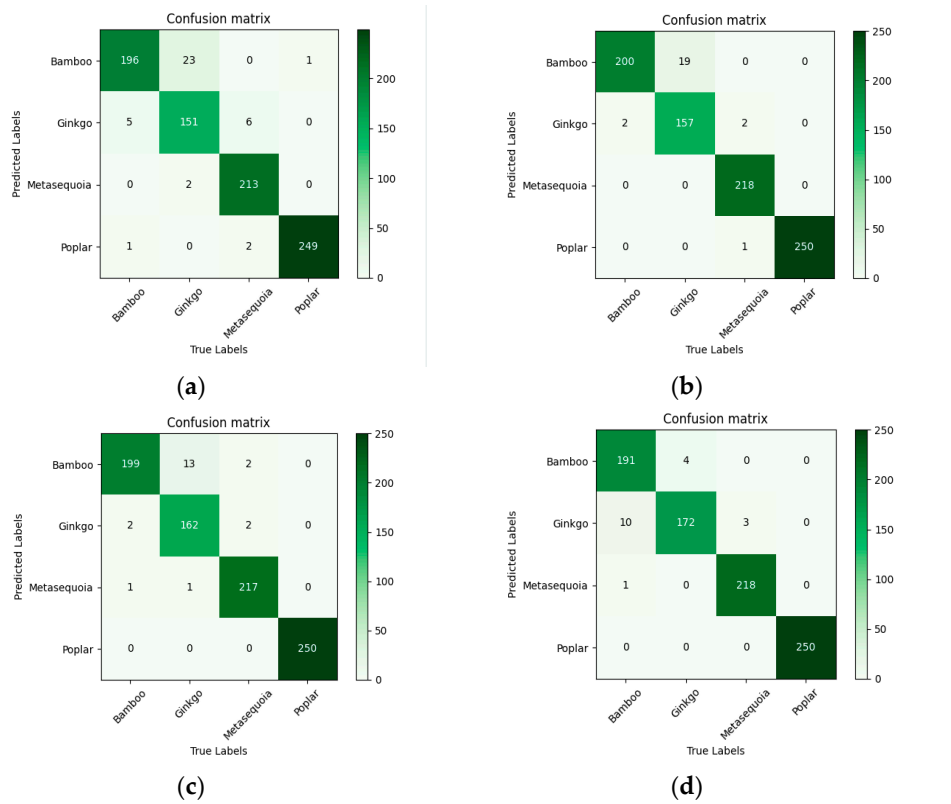

Figure 14. (**a**–**d**) are the confusion matrix plots for the ResNet-50, ConvNeXt-T, ViT-B, and Swin-T models validated on the original dataset, respectively.

The ResNet-50 model attained an OA of 95.32% with a Kappa coefficient of 0.9368. The ConvNeXt-T model reached an OA of 97.17% with a Kappa coefficient of 0.9621. The ViT-B and Swin-T models reached even higher OAs of 97.41% (with a Kappa coefficient of 0.9668) and 97.43% (with a Kappa coefficient of 0.9671), respectively. This study's finding showed that during training and validation, the Transformer models ViT-B and Swin-T outperformed the CNN model represented by ResNet-50 and ConvNeXt-T in terms of OA and Kappa coefficient.

This superiority of the Transformer models could be attributed to their multi-head attention mechanism, which allows them to capture richer feature information more effectively. The multi-head attention mechanism enables the models to attend to different regions of the input image simultaneously, capturing both local and global contextual information. In contrast, the CNN models, ResNet-50 and ConvNeXt-T, were found to be sensitive to factors such as blurry edges and weak texture in the original canopy images, which could impact the matching accuracy of feature points during feature extraction.

In summary, the research results show that the ViT-B and Swin-T models perform better than the ResNet-50 and ConvNeXt-T models in terms of OA and Kappa coefficient, due to the influence of factors such as blurry edges and weak textures in the original canopy images. This suggests that the ViT-B and Swin-T models have the potential to be effective in tree species classification tasks. However, an accurate judgment of model performance requires a comprehensive consideration of multiple evaluation metrics, as well as the needs of practical application scenarios, dataset characteristics, and model strengths and weaknesses. Furthermore, further research and empirical analysis can validate the performance of these models on different datasets and tasks.

4.2. Super-Resolution Reconstruction for Improved Tree Species Classification

Real-ESRGAN, a proven super-resolution image restoration algorithm [25], was utilized in this study to reconstruct the original image dataset. The reconstructed dataset was then used as input for training and validation in four different models. To ensure a fair comparison of the performance of the models, consistent hyperparameter tuning methods were applied during training.

Figure 15 provides a visual representation of the changes in the loss function and recognition accuracy for the training and validation sets of the reconstructed dataset for each of the four models. The figure allows for a detailed analysis of the training progress and stability of each model. It is observed that after approximately 90 to 100 iterations, the accuracy and loss of each model tend to stabilize, indicating convergence of the training process. This implies that the dataset reconstructed and repaired through super-resolution exhibits a faster convergence of the model with fewer training iterations compared to the original dataset, and demonstrates improved stability on the training set and validation set.

Furthermore, Figure 16 presents the OA, Kappa coefficient, and confusion matrix of each model. These metrics provide a comprehensive assessment of the performance of each model in terms of accuracy, agreement, and confusion among different classes. The detailed analysis of these metrics can provide insights into the effectiveness of each model in accurately classifying tree species based on the reconstructed dataset.

After comparing the model parameters reached and tree species classification results in Table 4, the following conclusions can be drawn: the ResNet-50 model reached an OA of 96.71% and a Kappa coefficient of 0.9558; the ConvNeXt-T model reached an OA of 98.70% and a Kappa coefficient of 0.9826; the ViT-B model reached an OA of 97.88% and a Kappa coefficient of 0.9716; and the Swin-T model attained an OA of 98.59% and a Kappa coefficient of 0.9810. Compared with the data samples that were not reconstructed using Real-ESRGAN, the recognition accuracy of each model increased by 1.39%, 1.53%, 0.47%, and 1.16%, respectively. Among them, the ConvNeXt-T model reached the best result. Therefore, we can conclude that the original image data, which may contain factors such as blurry edges and weak texture in the canopy, can benefit from reconstruction using

Real-ESRGAN. This reconstruction method can help improve the accuracy of tree species classification recognition to a certain extent.

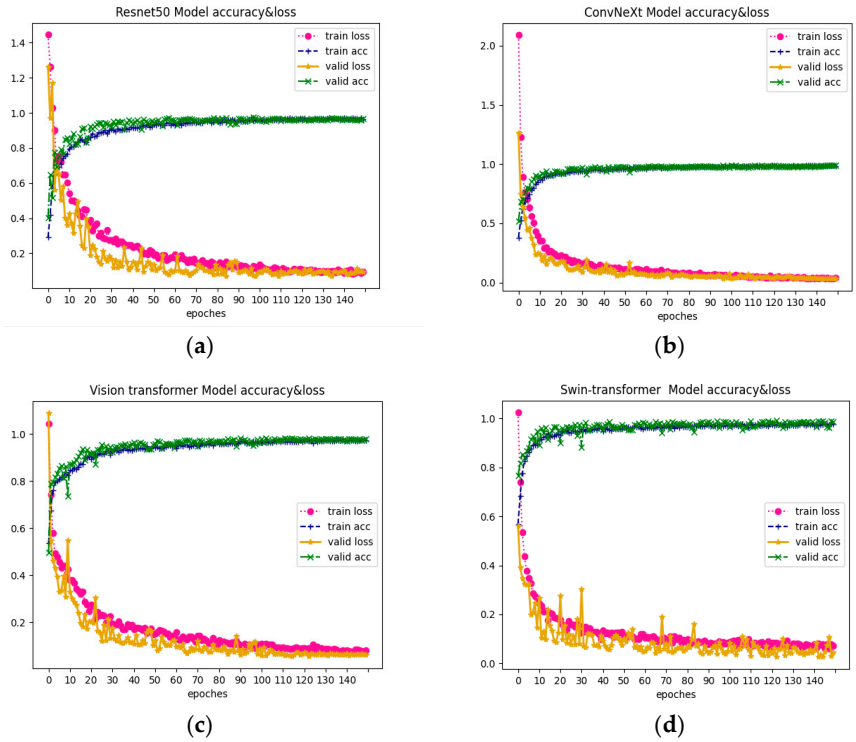

Figure 15. (**a**–**d**) depict the accuracy and loss rate of ResNet-50, ConvNeXt-T, ViT-B, and Swin-T models trained and validated on super-resolution reconstructed dataset, respectively.

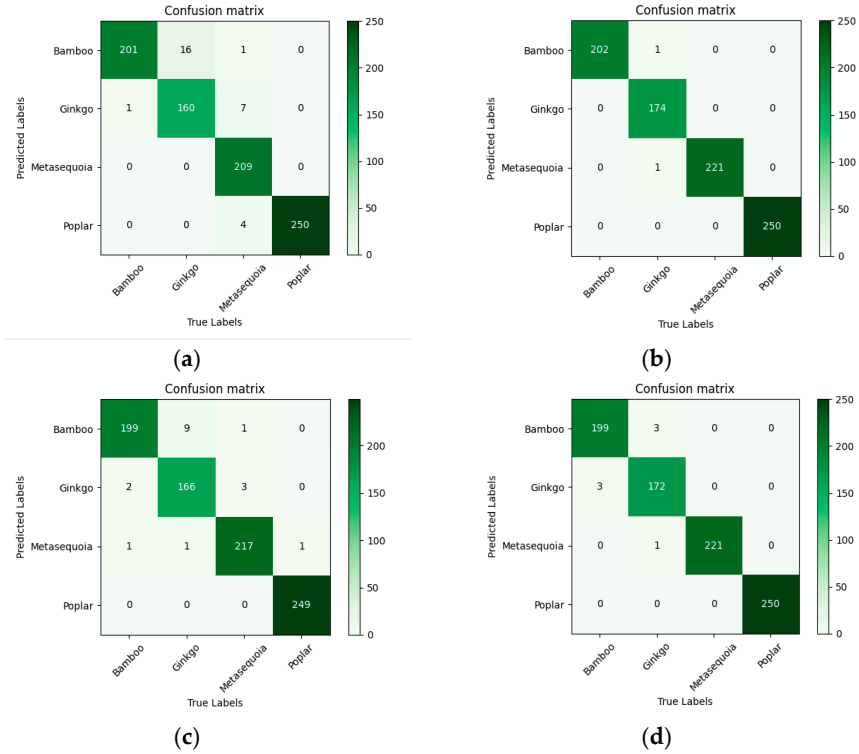

Figure 16. (**a**–**d**) depict the confusion matrix plots of ResNet-50, ConvNeXt-T, ViT-B, and Swin-T models validated on super-resolution reconstructed dataset, respectively.

Table 4. Comparison of classification results of different model tree species.

Model	Model Size/Mb Size	Average Accuracy of Model Validation for the Original Dataset/%	Average Accuracy of Model Validation after Super-Resolution Reconstruction Restoration of the Dataset/%
ResNet-50	25.55	95.32	96.71
ConvNeXt-T	28.58	97.17	98.70
ViT-B	86.41	97.41	97.88
Swin-T	28.26	97.43	98.59

4.3. Distribution Map of Tree Species in Dongtai Forest Plot

In this study, we selected a typical plot from the study area of Dongtai Forest Farm with UAV remote sensing images as a test sample set. To match the sample size of the original dataset, we used the same sliding window method to crop this sample. Then, we used the ConvNeXt-T model to test this sample set and created a tree species distribution map based on the model's predictions, as shown in Figure 17. The plot of this forest farm is divided as follows: the left plot is mainly planted with Metasequoia, the middle plot is mainly planted with Poplar, and the right plot is mainly distributed with Ginkgo. In addition, Bamboo mainly grows in the plot below.

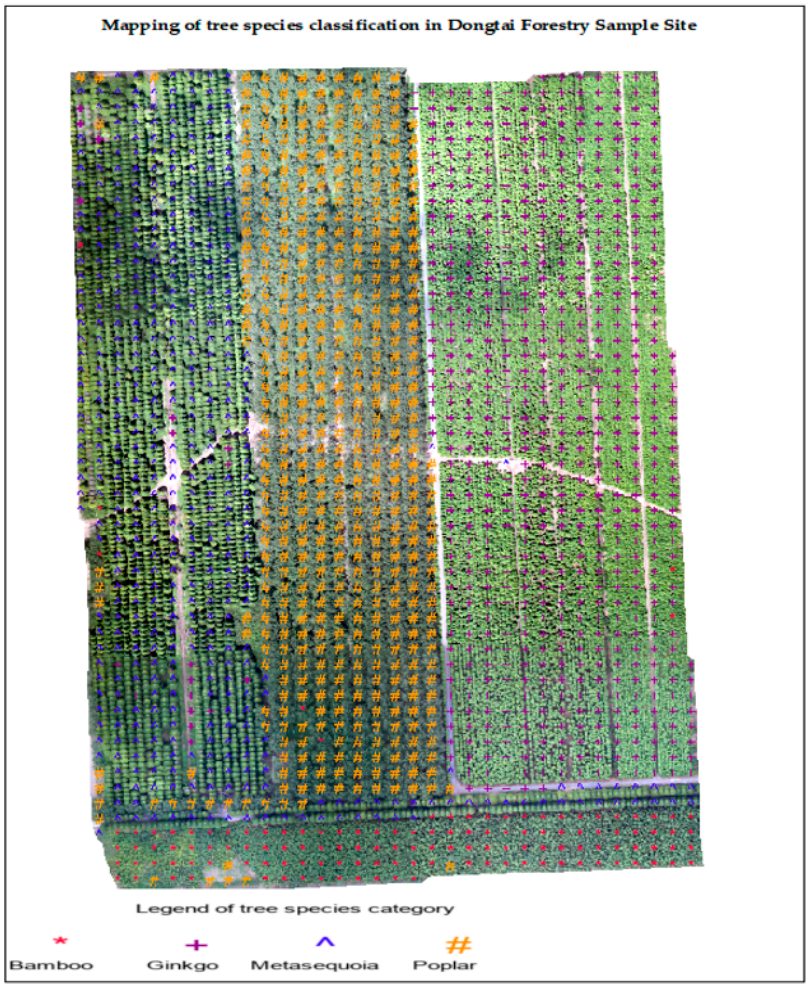

Figure 17. Tree species distribution map of Dongtai Forestry Sample Site.

This study examined the impact of different models on tree species classification in crown images. The study selected two representative pre-training models from the CNN and Transformer models, including ResNet-50, ConvNeXt-T, ViT-B, and Swin-T. The experimental results show that compared to traditional CNN models, Transformer models are more stable in feature extraction, and have better classification accuracy and stability. Additionally, this study used the Real-ESRGAN algorithm to perform super-resolution reconstruction and repair on the original image dataset, resulting in an improvement in the accuracy of tree species classification as demonstrated in the results. Finally, the study presents a distribution map of tree species in Dongtai Forest Farm, demonstrating the practical application of the Real-ESRGAN algorithm and serving as a reference for further research.

5. Discussion

5.1. Performance of CNN and Transformer in Classifying Tree Species Using the Original Dataset

For the application of tree species classification in low-altitude remote sensing images obtained from UAV, this paper further evaluated the classification accuracy performance of four models, namely, ResNet-50 and ConvNeXt-T as representatives of CNN models, and ViT-B and Swin-T as representatives of Transformer models, using the original canopy image dataset. Transformer has demonstrated its exceptional ability to capture global information, thereby bolstering a wide range of vision-related tasks such as image classification, object detection, and particularly semantic segmentation [33,34]. CNN and Transformer use object-based classification to achieve end-to-end tree species classification and avoid the non-transferability of manual feature extraction. The experimental results reveal that, as depicted in Figure 18, all four models exhibit classification validation accuracies exceeding 95%. Notably, the Swin Transformer reaches the highest classification accuracy, demonstrating an OA of 97.43% and a Kappa coefficient of 0.9671. Conversely, the CNN models, particularly the traditional CNN model, are more susceptible to the challenges posed by low-spatial-height aerial images, including detail loss, brightness reduction, blurred canopy edges, and weak image texture. These issues, coupled with small inter-class differences and significant intra-class differences, adversely impact the feature point matching accuracy of the CNN model, whereas the Transformer model is comparatively less affected.

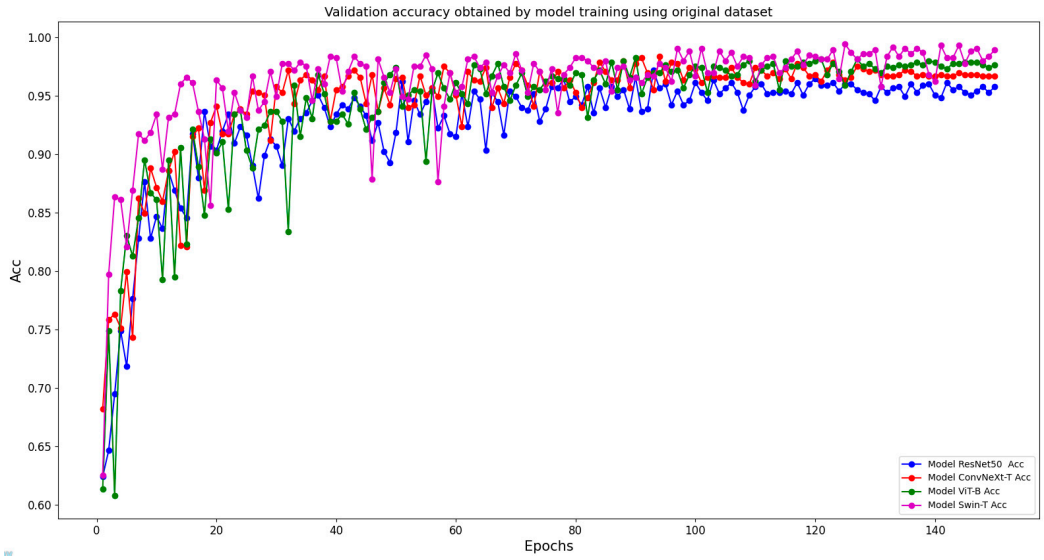

Figure 18. Validation accuracy obtained by training the model using the original dataset.

5.2. Performance of CNN and Transformer in Tree Species Classification Using Super-Resolution Reconstructed Dataset

To address these issues, this paper introduced the Real-ESRGAN super-resolution reconstruction technique to recover low-quality tree canopy images captured by UAVs.

The recovery process improved the validation accuracy of the four models. For example, the OA of the ConvNeXt-T model increased by 1.53% and the Kappa coefficient increased by 0.0205. The validation accuracy comparison derived from the model using the original dataset and the Real-ESRGAN processed dataset for training is depicted in Figure 19. Although the Real-ESRGAN super-resolution reconstruction technique has some limitations and shortcomings, it can be further improved in future research by introducing more fuzzy kernels and enhancing the image super-resolution algorithm model. These findings suggest that models trained on datasets restored and reconstructed by super-resolution may achieve stability faster while reaching higher accuracy on both training and validation sets compared to models trained on the original dataset. This phenomenon may be due to the fact that the restored and reconstructed datasets provide higher-quality images, which help the models to quickly acquire features related to tree species classification. However, further empirical evidence and validation are needed to confirm the correctness of this inference.

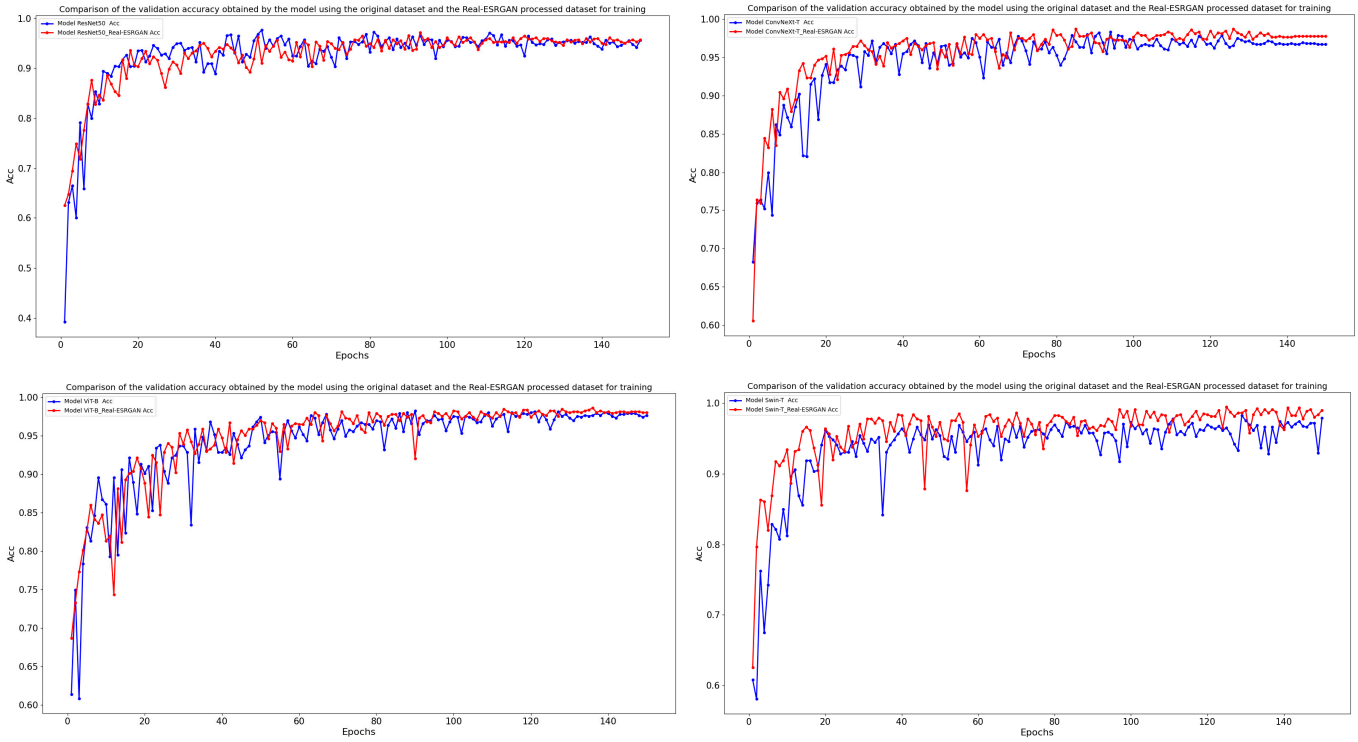

Figure 19. Comparison of the validation accuracy obtained by the model using the original dataset and the Real-ESRGAN processed dataset for training.

6. Conclusions

The assessment of tree species classification in Dongtai Forest utilizing RGB images captured by UAV has yielded promising outcomes through deep-learning-based approaches. Four models, including CNN models (ResNet-50 and ConvNeXt-T) and Transformer models (ViT-B and Swin-T), were trained and validated using UAV RGB tree crown images, achieving classification accuracies surpassing 95%. CNN models have been extensively used in forest resource surveys for tree species classification tasks, demonstrating exceptional classification accuracy [22,35]. Transformer models have also started finding applications in plant classification using UAV imagery [36] and exhibit significant potential for future advancements in forest surveys. However, the limited spatial resolution of aerial images introduces degradation challenges, such as detail loss, decreased brightness, blurred tree crown edges, and weak image texture. These issues negatively impact the feature point matching accuracy of CNN models and the capture of crucial information in the images. In contrast, Transformer models, with their inherent attention mechanisms, ef-

fectively leverage contextual information and global correlations in the images, resulting in comparatively less susceptibility to such issues. To address these challenges, Real-ESRGAN technology was adopted to perform super-resolution reconstruction and restoration on the original tree crown image dataset, leading to improved classification accuracy across all four models. This study confirms and underscores the observed enhancement in classification accuracy when using neural network models trained on images reconstructed through super-resolution. Super-resolution reconstruction techniques facilitate the restoration of low-quality images by recovering details, enhancing brightness, improving tree crown edge clarity, and augmenting image texture. These reconstructed images provide higher-quality information, enabling CNN and Transformer models to more accurately learn and extract features relevant to tree species classification. Consequently, when trained on these repaired and reconstructed image datasets, the four models exhibit improved stability and accuracy on validation sets.

Author Contributions: Conceptualization, Y.H. and X.W.; methodology, Y.H.; software, Y.H.; validation, Y.H., X.W. and Y.G.; formal analysis, Y.H.; investigation, Y.H.; resources, Y.H.; data curation, Y.H. and G.L.; writing—original draft preparation, Y.H.; writing—review and editing, Y.H., X.W., Y.G. and Y.Z.; visualization, Y.H.; supervision, X.W. and Y.G.; project administration, X.W.; funding acquisition, X.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Forestry Innovation Foundation of Guangdong Province (No. 2021KJCX001) and the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD). Additionally, financial support was provided by the Nanjing Institute of Environmental Sciences, MEE(GYZX230202).

Data Availability Statement: Data are available upon request from the section editors.

Acknowledgments: Y.H. is thankful for the patience and support from Jie Yang, who cooperated with him to write the paper in English and correct it and for the helpful discussion with Li Yang from the College of Forest, Nanjing Forestry University of China.

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

References

1. Zhao, D.; Pang, Y.; Liu, L.; Li, Z. Individual tree classification using airborne LiDAR and hyperspectral data in a natural mixed forest of northeast China. *Forests* **2020**, *11*, 303. [CrossRef]
2. Marrs, J.; Ni-Meister, W. Machine learning techniques for tree species classification using co-registered LiDAR and hyperspectral data. *Remote Sens.* **2019**, *11*, 819. [CrossRef]
3. Ballanti, L.; Blesius, L.; Hines, E.; Kruse, B. Tree species classification using hyperspectral imagery: A comparison of two classifiers. *Remote Sens.* **2016**, *8*, 445. [CrossRef]
4. Sun, Y.; Xin, Q.; Huang, J.; Huang, B.; Zhang, H. Characterizing tree species of a tropical wetland in southern china at the individual tree level based on convolutional neural network. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2019**, *12*, 4415–4425. [CrossRef]
5. Heikkinen, V.; Tokola, T.; Parkkinen, J.; Korpela, I.; Jaaskelainen, T. Simulated multispectral imagery for tree species classification using support vector machines. *IEEE Trans. Geosci. Remote Sens.* **2009**, *48*, 1355–1364. [CrossRef]
6. Zhang, Z.; Liu, X. Support vector machines for tree species identification using LiDAR-derived structure and intensity variables. *Geocarto Int.* **2013**, *28*, 364–378. [CrossRef]
7. Ab Majid, I.; Abd Latif, Z.; Adnan, N.A. Tree species classification using worldview-3 data. In Proceedings of the 2016 7th IEEE Control and System Graduate Research Colloquium (ICSGRC), Shah Alam, Malaysia, 8 August 2016; pp. 73–76. [CrossRef]
8. Bondarenko, A.; Aleksejeva, L.; Jumutc, V.; Borisov, A. Classification tree extraction from trained artificial neural networks. *Procedia Comput. Sci.* **2017**, *104*, 556–563. [CrossRef]
9. Raczko, E.; Zagajewski, B. Tree species classification of the UNESCO man and the biosphere Karkonoski National Park (Poland) using artificial neural networks and APEX hyperspectral images. *Remote Sens.* **2018**, *10*, 1111. [CrossRef]
10. Karlson, M.; Ostwald, M.; Reese, H.; Bazié, H.R.; Tankoano, B. Assessing the potential of multi-seasonal WorldView-2 imagery for mapping West African agroforestry tree species. *Int. J. Appl. Earth Obs. Geoinf.* **2016**, *50*, 80–88. [CrossRef]
11. Immitzer, M.; Atzberger, C.; Koukal, T. Tree species classification with random forest using very high spatial resolution 8-band WorldView-2 satellite data. *Remote Sens.* **2012**, *4*, 2661–2693. [CrossRef]
12. Hologa, R.; Scheffczyk, K.; Dreiser, C.; Gärtner, S. Tree species classification in a temperate mixed mountain forest landscape using random forest and multiple datasets. *Remote Sens.* **2021**, *13*, 4657. [CrossRef]

13. Burai, P.; Beko, L.; Lenart, C.; Tomor, T. Classification of energy tree species using support vector machines. In Proceedings of the 2014 6th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), Lausanne, Switzerland, 24–27 June 2014; pp. 1–4. [CrossRef]
14. da Rocha, S.J.S.S.; Torres, C.M.M.E.; Jacovine, L.A.G.; Leite, H.G.; Gelcer, E.M.; Neves, K.M.; Schettini, B.L.S.; Villanova, P.H.; da Silva, L.F.; Reis, L.P. Artificial neural networks: Modeling tree survival and mortality in the Atlantic Forest biome in Brazil. *Sci. Total Environ.* **2018**, *645*, 655–661. [CrossRef] [PubMed]
15. Freeman, E.A.; Moisen, G.G.; Frescino, T.S. Evaluating effectiveness of down-sampling for stratified designs and unbalanced prevalence in Random Forest models of tree species distributions in Nevada. *Ecol. Model.* **2012**, *233*, 1–10. [CrossRef]
16. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
17. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
18. Ling, C.; Jia, J.; Haiqing, W. An overview of applying high resolution remote sensing to natural resources survey. *Remote Sens. Nat. Resour.* **2019**, *31*, 1–7.
19. Nezami, S.; Khoramshahi, E.; Nevalainen, O.; Pölönen, I.; Honkavaara, E. Tree species classification of drone hyperspectral and RGB imagery with deep learning convolutional neural networks. *Remote Sens.* **2020**, *12*, 1070. [CrossRef]
20. Kapil, R.; Marvasti-Zadeh, S.M.; Goodsman, D.; Ray, N.; Erbilgin, N. Classification of Bark Beetle-Induced Forest Tree Mortality using Deep Learning. *arXiv* **2022**, arXiv:2207.07241. [CrossRef]
21. Hu, M.; Fen, H.; Yang, Y.; Xia, K.; Ren, L. Tree species identification based on the fusion of multiple deep learning models transfer learning. In Proceedings of the 2018 Chinese Automation Congress (CAC), Xi'an, China, 30 November–2 December 2018; pp. 2135–2140. [CrossRef]
22. Natesan, S.; Armenakis, C.; Vepakomma, U. Individual tree species identification using Dense Convolutional Network (DenseNet) on multitemporal RGB images from UAV. *J. Unmanned Veh. Syst.* **2020**, *8*, 310–333. [CrossRef]
23. Ford, D.J. UAV Imagery for Tree Species Classification in Hawai'i: A Comparison of MLC, RF, and CNN Supervised Classification. Ph.D. Thesis, University of Hawai'i at Manoa, Honolulu, HI, USA, 2020.
24. Chen, X.; Jiang, K.; Zhu, Y.; Wang, X.; Yun, T. Individual tree crown segmentation directly from UAV-borne LiDAR data using the PointNet of deep learning. *Forests* **2021**, *12*, 131. [CrossRef]
25. Wang, X.; Xie, L.; Dong, C.; Shan, Y. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Virtual, 11–17 October 2021; pp. 1905–1914.
26. Wang, X.; Yu, K.; Wu, S.; Gu, J.; Liu, Y.; Dong, C.; Qiao, Y.; Loy, C.C. ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks. In Proceedings of the European Conference on Computer Vision (ECCV) 2018, Munich, Germany, 8–14 September 2018; pp. 63–79. [CrossRef]
27. Schonfeld, E.; Schiele, B.; Khoreva, A. A u-net based discriminator for generative adversarial networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 8207–8216.
28. Yan, Y.; Liu, C.; Chen, C.; Sun, X.; Jin, L.; Peng, X.; Zhou, X. Fine-grained attention and feature-sharing generative adversarial networks for single image super-resolution. *IEEE Trans. Multimed.* **2021**, *24*, 1473–1487. [CrossRef]
29. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, Nevada, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
30. Liu, Z.; Mao, H.; Wu, C.Y.; Feichtenhofer, C.; Darrell, T.; Xie, S. A ConvNet for the 2020s. *arXiv* **2022**, arXiv:2201.03545. [CrossRef]
31. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *arXiv* **2017**, arXiv:1706.03762. [CrossRef]
32. Hendrycks, D.; Gimpel, K. Gaussian error linear units (gelus). *arXiv* **2016**, arXiv:1606.08415. [CrossRef]
33. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929. [CrossRef]
34. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Virtual, 11–17 October 2021; pp. 10012–10022. [CrossRef]
35. Egli, S.; Höpke, M. CNN-based tree species classification using high resolution RGB image data from automated UAV observations. *Remote Sens.* **2020**, *12*, 3892. [CrossRef]
36. Reedha, R.; Dericquebourg, E.; Canals, R.; Hafiane, A. Transformer neural network for weed and crop classification of high resolution UAV images. *Remote Sens.* **2022**, *14*, 592. [CrossRef]