remote sensing

MDPI

Article

FlexibleNet: A New Lightweight Convolutional Neural Network Model for Estimating Carbon Sequestration Qualitatively Using Remote Sensing

Mohamad M. Awad

Remote Sesning Center, National Council for Scientific Research, Beirut 11072260, Lebanon; mawad@cnrs.edu.lb

Abstract: Many heavy and lightweight convolutional neural networks (CNNs) require large datasets and parameter tuning. Moreover, they consume time and computer resources. A new lightweight model called FlexibleNet was created to overcome these obstacles. The new lightweight model is a CNN scaling-based model (width, depth, and resolution). Unlike the conventional practice, which arbitrarily scales these factors, FlexibleNet uniformly scales the network width, depth, and resolution with a set of fixed scaling coefficients. The new model was tested by qualitatively estimating sequestered carbon in the aboveground forest biomass from Sentinel-2 images. We also created three different sizes of training datasets. The new training datasets consisted of six qualitative categories (no carbon, very low, low, medium, high, and very high). The results showed that FlexibleNet was better or comparable to the other lightweight or heavy CNN models concerning the number of parameters and time requirements. Moreover, FlexibleNet had the highest accuracy compared to these CNN models. Finally, the FlexibleNet model showed robustness and low parameter tuning requirements when a small dataset was provided for training compared to other models.

Keywords: peri-urban forests; lightweight convolutional neural network; FlexibleNet; carbon sequestration; remote sensing

check for updates

Citation: Awad, M.M. FlexibleNet: A New Lightweight Convolutional Neural Network Model for Estimating Carbon Sequestration Qualitatively Using Remote Sensing. *Remote Sens.* **2023**, *15*, 272. https://doi.org/10.3390/rs15010272

Academic Editor: Dino Ienco

Received: 4 December 2022
Revised: 30 December 2022
Accepted: 31 December 2022
Published: 2 January 2023

1. Introduction

Since the advent of machine learning (ML) in the mid-twentieth century [1], it has played an important role in solving many complex problems such as image processing [2,3].

In the last decade, convolutional neural networks (CNNs), a sub-discipline of ML, have played an important role in advancing image processing such as segmentation, recognition, and classification sciences [4–7]. However, many networks suffered from huge computational resource and time requirements, such as ResNet50 [8], VGG16 [9], AlexNet [10], and GoogleNet [11]. Later, improvements to CNNs were introduced by reducing the number of layers and in turn reducing the number of parameters. The new generation of CNNs are called lightweight CNNs. The first lightweight model, SqueezeNet [12], showed classification accuracy close to AlexNet, and the number of parameters was only 1/510 compared to AlexNet. In addition to SqueezeNet, there are many lightweight models to mention, such as Xception [13], MobileNet [14], MobileNetV3 [15], ShuffleNet [16], and recently EfficientNet [17]. The last lightweight network has seven versions from B0 to B7.

However, some of these introduced lightweight CNN models still suffer from a growing amount of parameter tuning or inefficiency when there are insufficient samples [18]. Many researchers tried to improve some of these network models such as VGG16, ResNet50, and MobileNet by adding an auxiliary intermediate output structure named ElasticNet [19,20] that was directly connected to the network after each convolutional unit. Other researchers tried to improve the lightweight CNNs [21] by using MobileNet to extract deep and abstract image features. Each feature was then transformed into two features with two different convolutional layers. The transformed features were subjected to a Hadamard

product operation to obtain an enhanced bilinear feature. Finally, an attempt was made to improve lightweight CNNs by introducing a model called DFCANet [22] for corn disease identification. The model consisted of dual feature fusion with coordinate attention (CA) and downsampling (DS) modules. The CA module suppressed the background noise and focused on the diseased area. In addition, the DS module was used for downsampling. The above models enhanced the existing CNN models or solved specific problems.

Carbon is one of many greenhouse gases that exist naturally in the Earth's system [23]. However, carbon dioxide emissions have increased abnormally because of using fossil fuels for energy and due to land use/cover (LULC) changes. The fast increase in the carbon dioxide concentration in the air is making a major contribution to possible climate change and in turn to natural disasters as well as environmental and economic losses in the future [24]. The world's total forest area is about 4 billion hectares, corresponding to about 31% of the total land area [25]. Forests that include one or mixed types of trees with different plants absorb air pollution and provide the oxygen we breathe through photosynthesis, which absorbs carbon dioxide and preserves it in the leaves and stems up to the roots. Planted forests and woodlots were found to have the highest CO_2 removal rates, ranging from 4.5 to 40.7 t CO_2 ha^{-1} $year^{-1}$ during the first 20 years of growth [26,27].

Remote sensing data and methods are widely used to estimate carbon sequestration. Liu et al. [28] used airborne radar data to identify single-tree parameters such as the diameter at breast height (DBH) and tree height, and based on these measurements they estimated the AGB of single trees. Lizuka and Tateishi [29] used Landsat 8 and lso/Palsar to estimate forest tree volumes and tree ages. They used the extracted information to estimate carbon sequestration, and the verification was based on the collected field samples. Castro-Magnani et al. [30] used MODIS gross primary productivity (GPP) and net primary productivity (NPP) [31] to estimate carbon sequestration in the AGB. Later, they calculated the socio-economic benefit of sequestering carbon. Published research [32] has used airborne light detection and ranging (LiDAR) to acquire the vertical structure parameters of coniferous forests to construct two prediction models of aboveground carbon density (ACD). One is a plot-averaged height-based power model, and the other is a plot-averaged daisy-chain model. The correlation coefficients were significantly higher than that of the traditional percentile model. A paper published by Kanniah et al. [33] utilized different vegetation indices (Vis) and very high resolution WorldView-2 images to estimate carbon sequestration in an urban area. One of the Vis correlated strongly with the collected field data. However, the forest consisted of single tree species, which made the authors' research work simple. Uniyal et al. [34] estimated carbon sequestration using Landsat 8 and support vector machine (SVM) [35], random forest [36], k-nearest neighbor (kNN) [37], and the eXtreme gradient boosting (XGBoost) [38]. The authors used a huge number of variables extracted from Landsat image as inputs and field-collected data as training samples, and based on the R squared (coefficient of determination) they concluded that machine-learning-algorithm regressions are better than a linear regression. Zhang et al. [39] compared a convolutional neural network (CNN) to SVM and RF for estimating carbon sequestration in forests' AGB from Sentinel-2, Sentinel-1, and lso/Palsar. The authors used more than 67 variables to train the algorithms. The results showed that the CNN was better than RF and SVM at estimating carbon sequestered above the surface.

A literature review showed different attempts to estimate carbon sequestration using LIDAR data, which is limited by the technology's availability and cost and the size of the covered area. Some researchers used only one type of remote sensing optical data to extract vegetation indices (Vis) to compare some machine learning algorithms in estimating carbon sequestration. Other researchers used only optical images to calculate Vis and to estimate carbon sequestration in urban areas. Researchers deployed both optical and radar data without using machine learning to estimate carbon sequestration. One successful study combined multiple types of radar and optical data to compare machine learning algorithms, including a CNN, in estimating carbon sequestration in forests' AGB. However, this led to the need to calculate a large number of variables, and it demanded huge

computation resources. It is also known that a CNN alone is more effective in detecting patterns than estimating specific information [40,41]. Moreover, all the above research shared one objective, which was quantitatively estimating carbon sequestration by AGB.

The objectives and the contributions of this research are the following: (1) creating a new lightweight CNN model (FlexibleNet); (2) testing the new model (FlexibleNet) for qualitatively estimating carbon sequestration in peri-urban forests' AGB; and (3) creating new datasets that combine multispectral satellite images and multicriteria themes with different sizes. These datasets and python programs are available on GitHub.

Many issues make the new model better than other lightweight CNN models. First, the new model's flexibility arises from its ability to adapt to changes in tuning many parameters, such as the image dimension, dataset size, and layer depth and width. Second, the model uses only three extracted features from Sentinel-2 as inputs compared to the multi-input for other CNN models. Third, the new lightweight model can qualitatively measure carbon sequestration in peri-urban forests. Fourth, it is more efficient in dealing with small datasets.

After the introduction section, the second section describes the data, the third section contains the methods, the fourth section presents the experimental results, and the final section provides our conclusions.

2. Data

2.1. Area of Study and Field Survey

The border of the study area is specified by a red square in Figure 1. It is located in the El-Bared river basin in the northeast of Lebanon. The selection was based on many criteria that included the diversity of the forest types, forest densities, the existence of urban economic activities, the pressure exerted by the residents on the forest cover (cutting and burning), the ease of accessibility to the area (specific spots), and the existence of local authority support for fieldwork.

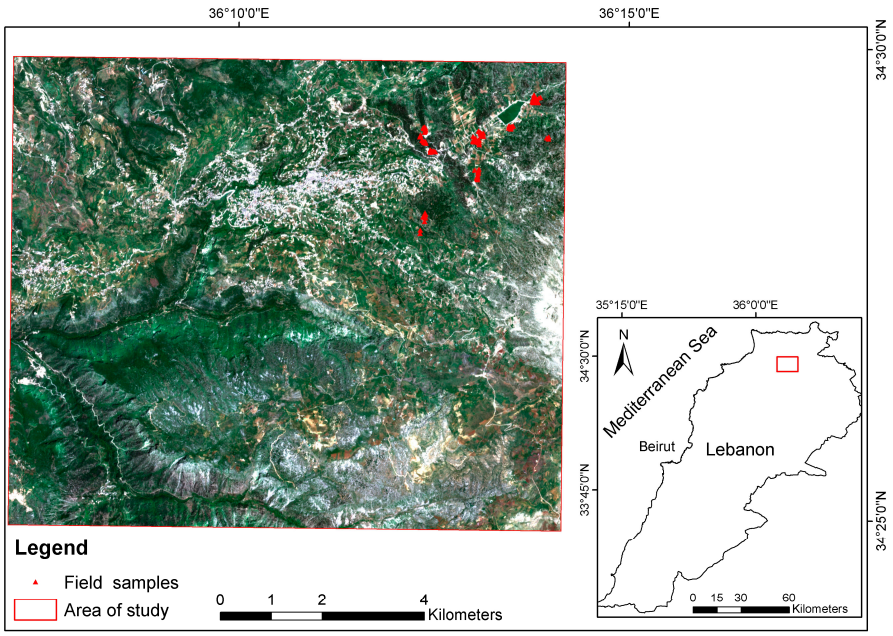

Figure 1. Study area.

The area of study occupies about 106.5 km^2 of different land cover types such as fruit trees, urban (including touristic facilities), forests, grasslands, etc. The highest elevation in the area of study is 1500 m, and the landform is flat to moderately steep (a slope less than 30%).

One can notice in Figure 1 that the field samples that were collected in the northeastern part of the study area. The selection of the field sampling area was based on having different forest types such as pine "*Pinus brutia*", cedar "*Cedrus Libani*", fir "*Abies cilicica*", juniper "*Juniperus excelsa*", and oak "*Quercus Cerris*" (Figure 2a–e).

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

Figure 2. Forest types: (**a**) *Quercus Cerris*, (**b**) *Abies cilicica*, (**c**) *Cedrus Libani*, (**d**) *Juniperus excelsa*, (**e**) *Pinus brutia*.

The sample collection was a random process, and it depended on the ease of accessibility to the investigated area. Table 1 shows the species type, the number of collected samples, the average height, and the average diameter at breast height (DBH). The cedars' cover was very small compared to other forest covers, and the authorities prohibited access to these trees because they were located in a reservation and they are national symbol.

Table 1. Information about the collected field samples.

Type	Number of Samples	Average DBH (cm)	Average Height (Meters)
Quercus Cerris	17	119	15
Pinus brutia	19	125	12
Abies cilicica	46	237	17
Juniperus excelsa	32	225	8

2.2. Data Type and Source

In this research, we deployed Sentinel-2 data, which is considered to be important and free optical remote sensing satellite data. Sentinel-2A and Sentinel-2B were launched in June 2015 and March 2017, respectively [42]. Sentinel-2 is an optical remote sensing satellite. It has a spatial resolution that varies between 10 m and 60 m depending on the wavelength. Sentinel-2A has a temporal resolution of 10 days, which can become 5 days with the combination of Sentinel-2B and another optical satellite with the same specifications as Sentinel-2A. The clipped image has a size of 1115 × 955 pixels and consists of bands 3, 4, and 8, which correspond to green, red, and near infrared. These bands were selected for two reasons: they have the highest spatial resolution, and they are representative of the crops' photosynthesis process. To extract the required area, we used Google Earth Engine's (GEE) Sentinel-2 dataset and computation facilities. One Sentinel-2 image was selected in May 2020 for two reasons: to reduce the cloud cover effect (less than 5% of the image size) and to obtain the maximum vegetation cover (deciduous and coniferous trees, grasslands, and agricultural lands).

Moreover, a vector layer representing the global canopy height for the year 2020 at a 10 m resolution [43] was used in the canopy density model (Figure 3).

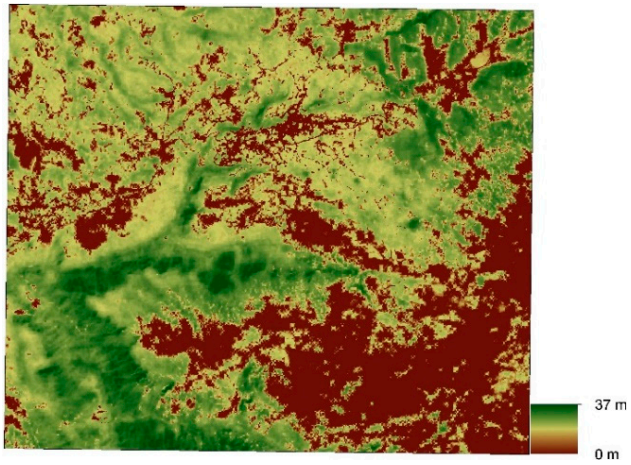

Figure 3. Canopy height map.

3. Methods

The following flowchart (Figure 4) shows the different tasks that were implemented in this research to qualitatively estimate carbon sequestration in ABG forests using the new lightweight CNN model (FlexibleNet) and the training and Sentinel-2 image datasets.

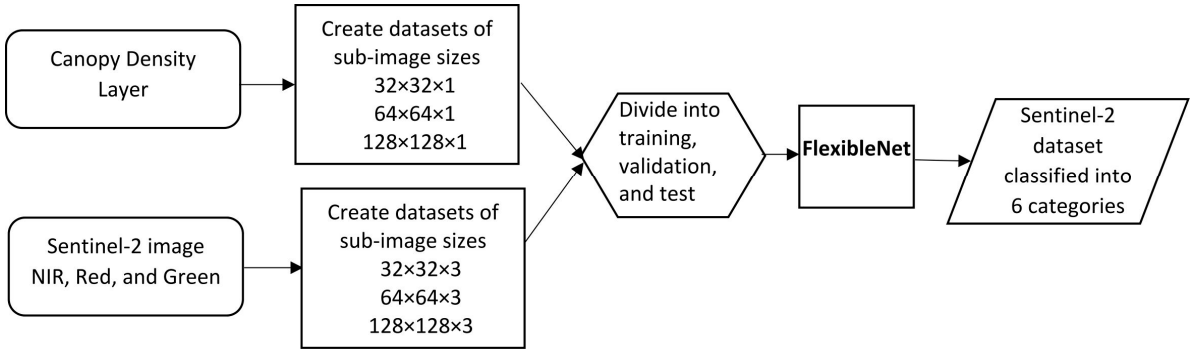

Figure 4. The general process for qualitatively assessing forests' AGB carbon sequestration capacities.

3.1. Canopy Density Model (CDM)

An adapted model created by Abdollahnejad et al. [44] incorporated different indices from Sentinel-2 images and the thermal band of Landsat to create a canopy density model. The adapted model combined different resolutions, which lowered the credibility and efficiency of the final product. Moreover, the model neglected the canopy heights, which can successfully differentiate between forests and other vegetation types.

Both the Sentinel-2 image (level 2) and the canopy height layer were obtained using the Google Earth Engine (GEE) platform. Scripts were written in the Java language to retrieve the needed data. Normally, the acquired Sentinel-2 image is level 2, which is an image that is corrected geometrically and atmospherically. Three indices were created from the Sentinel-2 image using the following equations:

$$AVI = [(NIR + 1) \times (1 - Red) \times (NIR - Red)]^{1/3} \quad (1)$$

$$BI = \frac{(NIR + Green) - Red}{(NIR + Green) + Red} \quad (2)$$

$$SI = \sqrt{(1 - Green) \times (1 - Red)} \quad (3)$$

where *AVI* is the advanced vegetation index, *BI* is the bare soil index, and *SI* is the canopy shadow index. Moreover, *NIR*, *Red*, and *Green* represent the three different spectrums and

the bands B2, B3, and B8 in the Sentinel-2 image. *AVI* was modified to provide values between −1 and 1. The modification included replacing 256 with 1 and normalizing the bands. *BI* ranged between 0 and 1, where 0 meant complete bare soil or no vegetated area and 1 meant completely covered by vegetation. Finally, *SI* was modified by replacing 256 with 1, and the bands were normalized. *SI* values ranged between 0 and 1, where the maximum value indicated the highest canopy shadow.

These themes, including the canopy heights, were classified into six categories using natural break classification (Jenks) [45]. The classes were based on natural groupings inherent in the data. Normally, the classification process identifies breakpoints by picking the class breaks that best group similar values and maximize the differences between classes. Finally, a spatial analysis that included mathematical operations was deployed to obtain the canopy density theme. The above processes were combined according to the following flowchart (Figure 5).

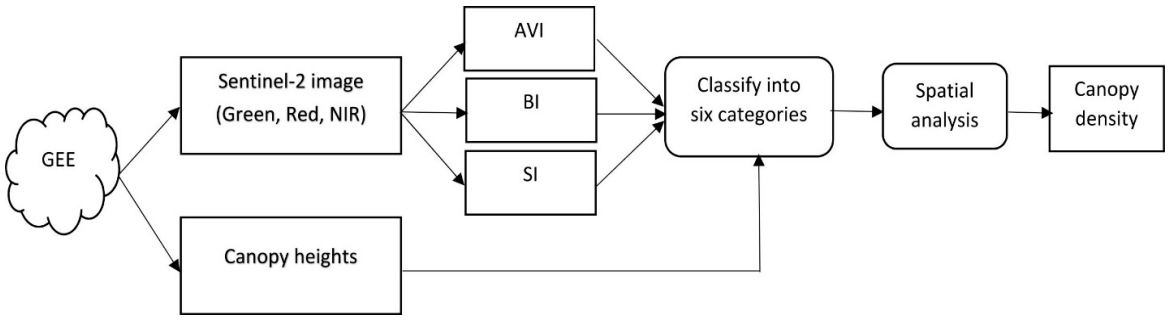

Figure 5. Canopy density model.

Further investigation in the future to improve the canopy density layer may include higher-spatial-resolution satellite images and time series of NDVI to separate deciduous forest trees from evergreens, which could further enhance research work.

3.2. The New Lightweight Convolutional Neural Network Model (FlexibleNet)

CNNs are collections of neurons that are ordered in inter-related layers, with convolutional, pooling, and fully connected layers [46]. CNNs require less preprocessing, and they are the most effective learning algorithms for realizing image structures. Moreover, it was proven that CNNs excel in image classification, recognition, and retrieval [47].

Normally, a simple CNN model consists of one or many of the following layers: 1—convolutional layer, 2—pooling layer, 3—activation layer, and a fully connected layer.

In this research, we created a new lightweight CNN model (FlexibleNet) to reduce the resource and training dataset requirements (Figure 6). The performance of the new model was tested in the qualitative classification of carbon sequestration. Our new model is a CNN scaling-based model (width, depth, and resolution). The depth corresponds to the number of layers in a network. The width is associated with the number of neurons in a layer or, more pertinently, the number of filters in a convolutional layer. The resolution is simply the height and width of the input image. Unlike the conventional practice, which arbitrarily scales these factors, FlexibleNet uniformly scales the network width, depth, and resolution with a set of fixed scaling coefficients.

We combined different strategies to improve the FlexibleNet performance. These strategies were spatial exploitation and varying the depth. Spatial exploitation includes parameters such as the number of processing units (neurons), filter size, and activation function. We assumed that varying the CNN's depth can better approximate the target function with a number and can improve feature representations and network performance.

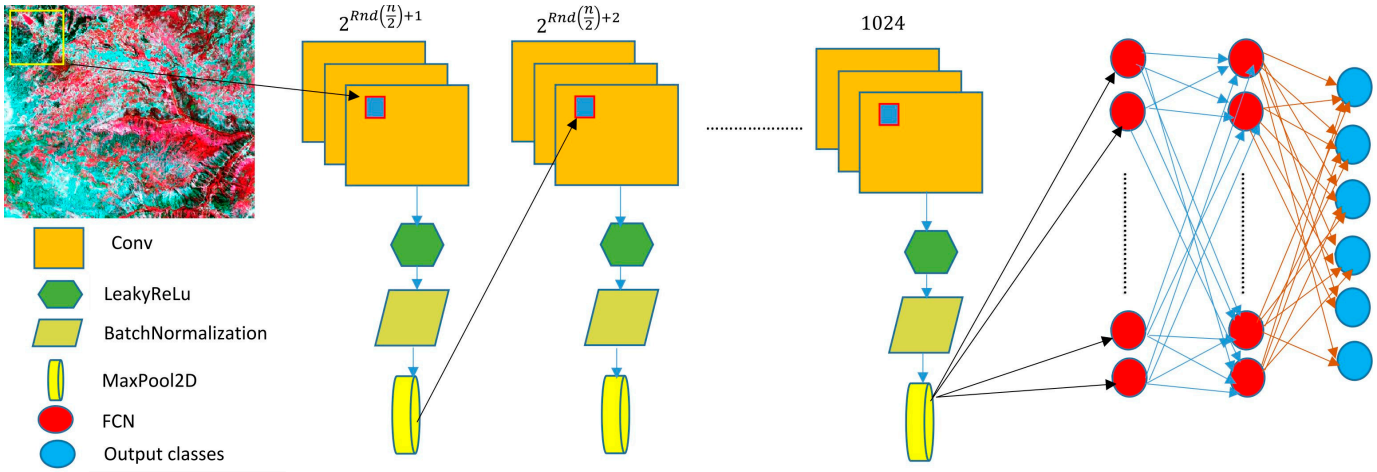

Figure 6. The inner structure of the FlexibleNet model.

The spatial exploitation included changes to the filter size and the activation function. Moreover, the depth of the FlexibleNet network or the number of convolutional layers varied when the dimensions of the features changed. The variations in the width and depth were based on the variation in the resolution of the image. The following equations depict the changes to filter size:

$$Im = \sum_{i=1}^{m} d_i \times w_i \times c_i \quad \rightarrow d = w = 2^n \quad \rightarrow f = Rnd\left(\frac{n}{2}\right) \times Rnd\left(\frac{n}{2}\right) \tag{4}$$

where Im is the original image and m is the number of sub-images of size $d \times w \times c$, where d is the number of rows, w is the number of columns, and c is the number of channels. Moreover, n is the exponent, f is the filter size, and $Rnd()$ is the round function (d, w, c, and $n \in \mathbb{Z}^+$). If the image (Im) has an uneven size, zeros are padded to the columns and/or rows to make them even.

The number of filters for each convolution layer can be set up based on the following rules:

$$\begin{cases} f_m = & \begin{matrix} Initial\ 2^{Rnd(\frac{n}{2})+1}\ where\ n \geq 5 \\ m = Rnd\left(\frac{n}{2}\right) + 2; \quad 2^m \ \rightarrow m = m + 1 \\ Final \quad 1024 \end{matrix} \end{cases} \tag{5}$$

where f_m represents the filter sizes. These rules work as follows: Suppose I have a sub-image of size 32 × 32. Then, n = 5. This means that the initial filter is f_0 = 16. Next, the filter size is obtained by calculating m f_m = 32, 64, 128, 256, and 512, where m = 6, 7, 8, and 9 and the final filter size is 1024 (maximum) with m = 10.

Then, the leaky rectified linear activation function (LReLU) is used [48], which is a modification of the ReLU activation function. It has the same form as the ReLU, but it will leak some positive values to 0 if they are close enough to zero (Equation (6)). It is a variant of the ReLU activation function. Normally, ReLU is half-rectified (from the bottom). ReLU(p) is zero when p is less than zero, and ReLU(p) is equal to p when p is above or equal to zero.

$$LReLU(p) = \max(0.01 \times p, p) \tag{6}$$

The number of layers or the depth of the network (Lay_{depth}) can be computed as indicated in Equation (7). It is noticeable that the depth reached unity when the dimensions of the image were >18. The creation of Equation (7) was based on the assumptions that a sub-image cannot be less than 16 × 16 and that the maximum sub-image size is the image itself. Adapting to the increase in the sub-image size requires decreasing the network depth by one level (the number of convolution layers) each time the sub-image increases. The

depth starts from $Rnd\left(\frac{n}{2}\right) + 7$ convolution layers to one layer, where the size of the image is the image itself, assuming it may reach infinity as a size.

$$Lay_{depth} = \begin{cases} Rnd\left(\frac{n}{2}\right) + 7 & 4 \leq n \\ Rnd\left(\frac{n}{2}\right) + 5 & 5 \leq n \leq 6 \\ Rnd\left(\frac{n}{2}\right) + 3 & 7 \leq n \leq 8 \\ Rnd\left(\frac{n}{2}\right) + 1 & 9 \leq n \leq 10 \\ Rnd\left(\frac{n}{2}\right) - 1 & 11 \leq n \leq 12 \\ Rnd\left(\frac{n}{2}\right) - 3 & 13 \leq n \leq 14 \\ Rnd\left(\frac{n}{2}\right) - 5 & 15 \leq n \leq 16 \\ Rnd\left(\frac{n}{2}\right) - 7 & 17 \leq n \leq 18 \\ 1 & n > 18 \end{cases} \quad (7)$$

In addition to the Lay_{depth} size, there is a fixed number of three dense layers (DL). According to [49], the dense layer is an often-used layer that contains a deeply connected neural network layer. DL is a hidden layer associated with one node in the next layer.

Figure 7a–c show the FlexibleNet structure for three different scales based on the above rules, where n = 32, 256, and 512. One can notice that as the scale increases, the depth decreases. This strategy can help reduce the computation requirements (processing power and memory size).

3.3. Estimating Carbon Sequestration for the Collected AGB Samples

The measured trees were used to compute the volume of the AGB using Equations (8) and (9). Where Vm^3 is the volume of wood in cubic meters, Hm is the height of the tree, DBH is the diameter at breast height, and Bm^2 is the base area in square meters. Lee et al. [50] suggested Table 2 to help in the calculation process of carbon sequestration in the ABG. The carbon content usually uses a value of 0.5, which means that wood is about 50% carbon. We used the model created by Lizuka and Tateishi [29] to estimate carbon sequestration per hectare (CS_{ha}) (Equation (10)). Fc = 44/12 converts the carbon value to the carbon dioxide sequestration value, where 12 and 44 represent the molecular masses of carbon and carbon dioxide, respectively.

$$Vm^3 = Bm^2 \times Hm \quad (8)$$

$$Bm^2 = \pi \times \left(\frac{DBH}{2}\right)^2 \quad (9)$$

$$CS_{ha} = Vm^3 \times Be \times Bd \times Cc \times Fc \quad (10)$$

Table 2. Coefficients for calculating carbon sequestration by forest type.

Type of Forest	Bulk Density (Bd) (Tons/m^3)	Biomass Expansion (Be)	Carbon Content (Cc)
Coniferous	0.47	1.651	0.5
Deciduous	0.80	1.720	0.5
Mixed	0.635	1.685	0.5

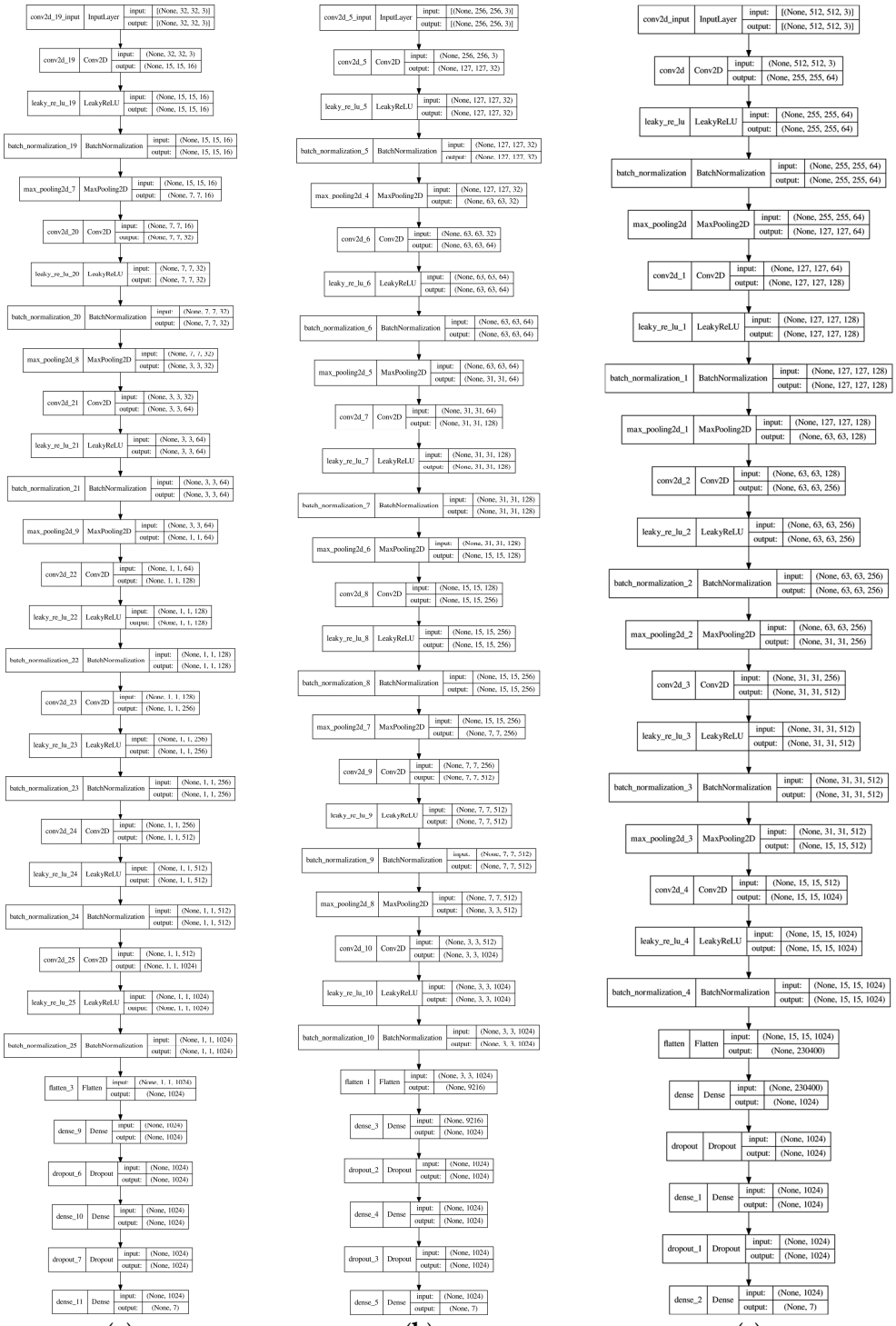

Figure 7. FlexibleNet with different scales and depths: (**a**) 32, 7; (**b**) 256, 6; (**c**) 512, 5.

4. Results

For this section, we created different datasets of Sentinel-2 sub-images to prove the efficiency of the new lightweight CNN model (FlexibleNet) in qualitatively estimating carbon dioxide sequestration. The collected samples of trees' characteristics, as shown in Table 1, were used to calculate CS_{ha} using Equations (8)–(10). Then, these values were converted to qualitative values using Sturges' rule [51]. Since the samples represent trees' characteristics, the "no carbon" class was removed. Figure 8 represents the distribution of the samples according to five classes (very low, low, moderate, high, and very high). These

created qualitative samples were used to verify the credibility of the canopy density dataset using a confusion matrix [52] before using it in the training of the new model (Table 3). The accuracy computed from the matrix using estimated versus measured values was 92.1%.

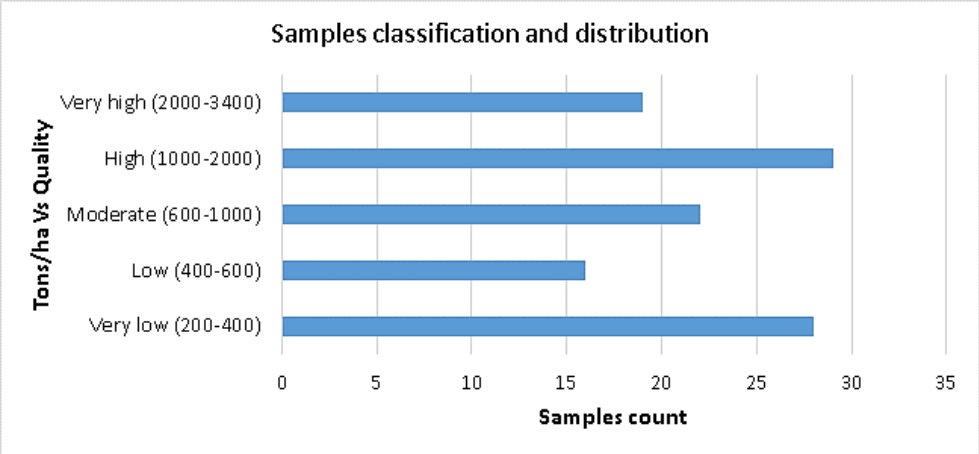

Figure 8. Trees samples classification.

Table 3. Confusion matrix.

Measured/Estimated	Very Low	Low	Moderate	High	Very High
Very low	23	2	1	2	0
Low	1	15	0	0	0
Moderate	0	1	21	0	0
High	1	1	0	27	0
Very high	0	0	0	0	19

We created different datasets that consisted of tiled sub-images with three different sizes of 32 × 32 (1050 images), 64 × 64 (270 images), and 128 × 128 (72 images) and three bands representing different spectrums (green, red, and near infrared). The other datasets consisted of the same size and number of tiles but only represented canopy densities with six classes (no carbon, very low, low, moderate, high, and very high). A script was written in the Python language to classify the Sentinel-2 sub-images into six classes based on the computed canopy density statistics (Algorithm 1). The script takes every computed sum (arr) for each canopy density sub-image and compares it to the created criteria (criteria) based on Sturges' rule.

The sums of the pixel values of all canopy density sub-images were calculated. Next, these sums' maximum, minimum, and average were computed. Then, they were used with Struges' rule to classify the Sentinel-2 sub-images into six classes. After that, the datasets were split into 80% training and 20% validation samples. Figure 9 shows examples of the original Sentinel 2 sub-images (false color) and their counterpart canopy density classes. The colors in the canopy density images signify that very low is dark brown, low is light brown, moderate is light green, high is green, and very high is dark green.

These samples were used as part of the training and validation datasets to check the efficiency of the FlexibleNet model.

Algorithm 1 A python script to classify Sentinel-2 sub-images.

```
if( (arr[i] ≤ 0) and file_exists):
# it checks if the sum is less or equal to zero and if the image exists in the folder
before copying it to the no carbon folder
shutil.copy(filename,dest1)
# copy image to no carbon folder (dest1)
if((arr[i] >0 and arr[i] ≤ criteria *2) and file_exists):
shutil.copy(filename,dest2)
# copy to folder very low
if((arr[i] > criteria *2 and arr[i] ≤ criteria *3) and file_exists):
shutil.copy(filename,dest3)
# copy to folder low
if((arr[i] > criteria *3 and arr[i] ≤ criteria *4) and file_exists):
shutil.copy(filename,dest4)
# copy to folder moderate
if((arr[i] > criteria *4 and arr[i] ≤ criteria *5) and file_exists):
shutil.copy(filename,dest5)
# copy to folder high
if((arr[i] > criteria *5 and arr[i] ≤ maxval) and file_exists):
shutil.copy(filename,dest6)
# copy to folder very high
```

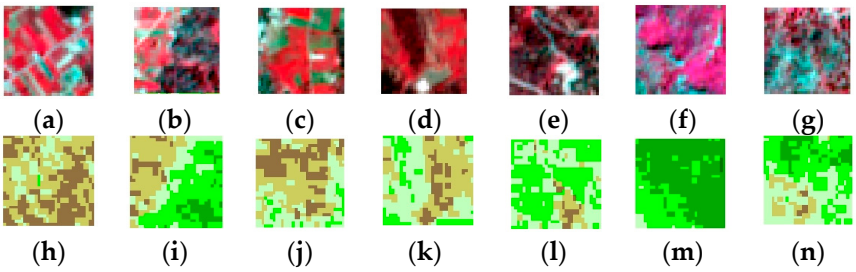

Figure 9. Different sub-images showing (**a**–**g**) the original Sentinel-2 images and (**h**–**n**) canopy density (very low, low, medium, high, and very high).

The FlexibleNet was compared with four popular and well-known convolutional neural networks: the large model ResNet50 [8], the lightweight models Xception [13] and MobileNetV3-Large [15], and the EfficientNet [17]. These models were selected based on their popularity, efficiency, and availability.

All the models, including FlexibleNet, were run using "Jupyter Notebook" on Amazon SageMaker cloud computing facilities that had 16 GB of memory capacity and two Intel Xeon Scalable processors with 3.3 GHz speed. Moreover, these models were run for a maximum of 100 epochs, and each epoch had several steps (number of steps per epoch = (total number of training samples)/batch size). We deployed a stochastic gradient descent (SGD) optimizer in FlexibleNet with an initial learning rate of 0.001. SGD is an iterative method for optimizing an objective function with suitable smoothness properties. SGD replaces the actual gradient (calculated from the entire dataset) with an estimate thereof (calculated from a randomly selected subset of the data). Especially in the high-dimensional optimization problem, this reduces the very high computational burden, achieving faster iterations in return for a lower convergence rate [53]. The learning rate of 0.001 was selected based on previous research conducted by Asif et al. [54].

In the first experiment, the datasets of 32 × 32 were used to compare these models. The outcomes of these models are shown in Table 4, and the behaviors of these models during the run process are shown in Figure 10a–j.

Table 4. Summary of the outcomes of the experiments using an image resolution of 32 × 32.

Model Name	Number of Parameters (Millions)	Time Requirement (Minutes)	Accuracy %	Lowest Loss Value
FlexibleNet	5.52	13.3	98.81	0.042
ResNet50	26.38	77	96.41	0.1074
EfficientNetB5	31.30	28.4	52	1.1
MobileNetV3-Large	6.23	13.3	68.69	0.7122
Xception	21.58	62	66.96	0.83

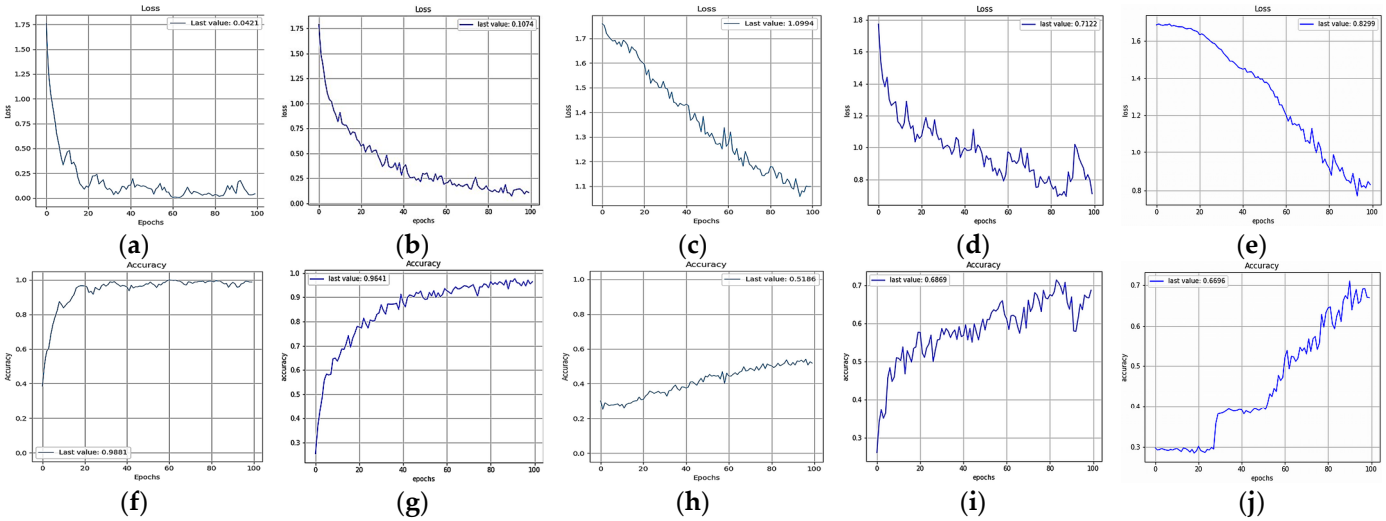

Figure 10. Loss and accuracy of sub-images of size 32 × 32 processed by (**a**,**f**) FlexibleNet, (**b**,**g**) ResNet50, (**c**,**h**) EfficientNetB5, (**d**,**i**) MobileNetV3-Large, and (**e**,**j**) Xception.

The new model had total parameters equal to 5.52 million. The total time was 13.3 min, with 8 s for each iteration. The accuracy of the final trained model was about 98.81%, and the final loss was 0.042.

The ResNet50 model was run for 100 iterations (epochs), with a total number of parameters equal to 26.38 million. ResNet50 took 77 min, with an accuracy of about 96.41%, and the final loss was 0.1074. The accuracy of ResNet50 was lower than that of FlexibleNet. This proved the reliability and efficiency of the new model.

EfficientNet was also tested using the same datasets. The number of iterations was 100, and the number of parameters was 31.3 million. It took the model 88.4 min to complete the iterations (epochs). The lowest loss was 1.1, and the highest accuracy was 52%. This proved that FlexibleNet is more efficient and accurate than the lightweight EfficientNet.

The lightweight network models MobileNetV3-Large and MobileNetV3-Small are normally targeted for high- and low-resource use cases. These models are then adapted and applied to object detection and semantic segmentation. MobileNetV3-Small is more suitable for mobile phone operating systems. MobileNetV3-Large is 3.2% more accurate in ImageNet classification while reducing latency by 15% compared to MobileNetV2 [55]. The implemented MobileNetV3-Large had 6.23 million total parameters, and it was run for 100 iterations. It took the model 13.3 min to complete the iterations (epochs). The lowest loss was 0.7122, and the highest accuracy was 68.69%. This proved that FlexibleNet was more efficient and accurate than the lightweight MobileNetV3-Large.

The second experiment was conducted in the same area of study, but the datasets had an image resolution of 64 × 64 pixels. Figure 11 shows different 64 × 64 sub-images alongside corresponding canopy sub-images. We placed constraints on running the FlexibleNet

model and the other tested models to avoid falling into the overfitting problem because of the lack of a large dataset of images [56].

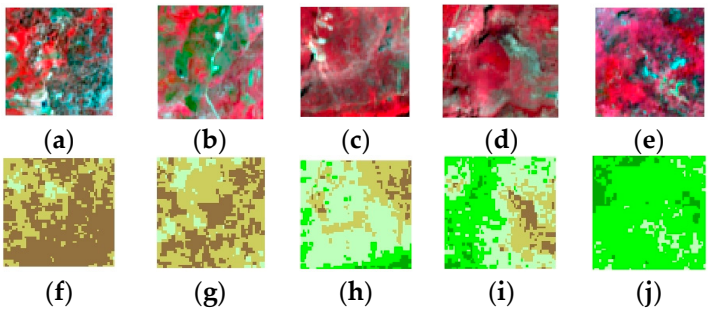

Figure 11. Sentinel-2 sub-images (64 × 64): (**a**–**e**) original and (**f**–**j**) canopy density (very low (dark brown) to very high (dark green)).

During the fitting process of these models, the loss function was tested. The fitting process was terminated when several iterations completed and the minimum loss value did not change. Table 5 shows the outcomes of testing the different models on different image resolutions. First, it is noticeable that the number of parameters increased for FlexibleNet, ResNet50, and EfficientNetB5. Nevertheless, the time requirement decreased for all models except EfficientNetB5. Finally, FlexibleNet was the only model with the highest accuracy and the lowest loss function value, as shown in Figure 12a–j.

Table 5. Summary of the outcomes of the experiments using an image resolution of 64 × 64.

Model Name	Number of Parameters (Millions)	Time Requirement (Minutes)	Accuracy %	Lowest Loss Value	Total Iterations
FlexibleNet	8.4	5	98.25	0.0457	60
ResNet50	32.6	13	96.74	0.0877	51
EfficientNetB5	32.9	40	93.06	0.1936	100
MobileNetV3-Large	6.23	4	90.22	0.3422	74
Xception	21.58	22	31.52	1.718	100

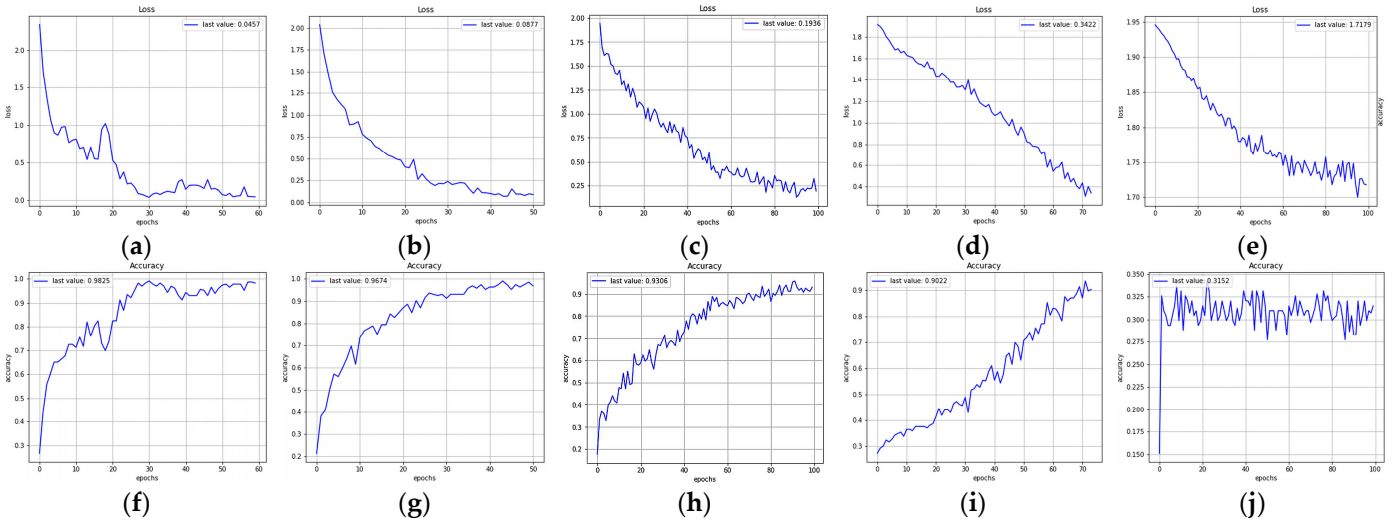

Figure 12. Loss and accuracy of 64 × 64 sub-images processed by (**a**,**f**) FlexibleNet, (**b**,**g**) ResNet50, (**c**,**h**) EfficientNetB5, (**d**,**i**) MobileNetV3-Large, and (**e**,**j**) Xception.

The final experiment was conducted in the same area of study, but the image resolution was 128 × 128 pixels, which resulted in smaller datasets. Figure 13 shows different 128 × 128 sub-images alongside corresponding canopy density sub-images.

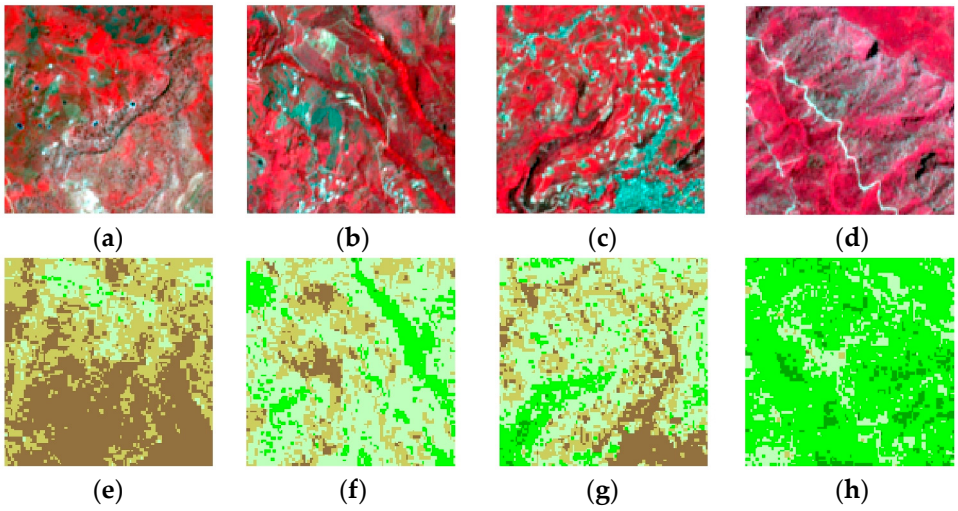

Figure 13. Sentinel-2 sub-images (128 × 128): (**a**–**d**) original and (**e**–**h**) canopy density.

We placed constraints on running the FlexibleNet model and the other tested models to avoid falling into the overfitting problem because of a lack of a large dataset of images [56]. The loss function was tested, and the fitting process was terminated when the minimum loss value did not change after a specific number of iterations.

Table 6 lists the results of running different models. Again, FlexibleNet and MobileNetV3-Large showed stable numbers of parameters, even when the dimensions of the image increased from 64 × 64 to 128 × 128. However, FlexibleNet was the fastest, and it had the highest accuracy and lowest loss value compared to the other models. In this experiment, FlexibleNet showed robustness in dealing with very small datasets (72 images), whereas the others failed to deal with the problem. Many adjustments were made (such as duplicating the dataset) to overcome the limited size of the dataset and make the other models run smoothly. The performances of these models (accuracy and loss) are shown in Figure 14a–j.

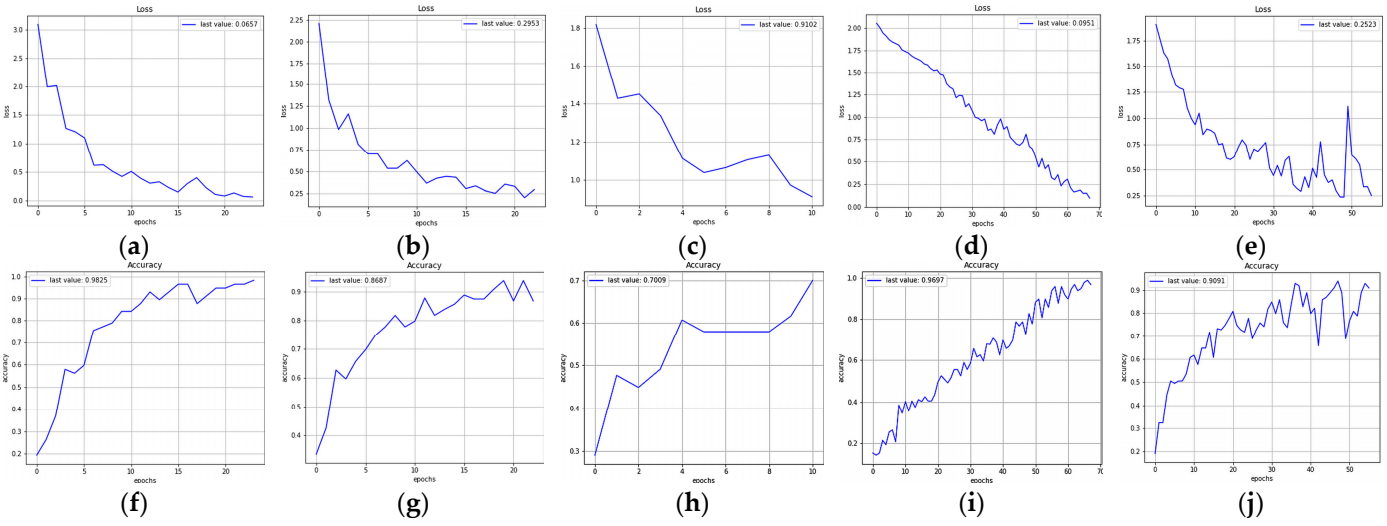

Figure 14. Loss and accuracy of 128 × 128 sub-images processed by (**a**,**f**) FlexibleNet, (**b**,**g**) ResNet50, (**c**,**h**) EfficientNetB5, (**d**,**i**) MobileNetV3-Large, and (**e**,**j**) Xception.

Table 6. Summary of the outcomes of the experiments using an image resolution of 128 × 128.

Model Name	Number of Parameters (Millions)	Time Requirement (Minutes)	Accuracy %	Lowest Loss Value	Total Iterations
FlexibleNet	8.4	0.8	98.25	0.0657	24
ResNet50	57.8	7.7	86.87	0.2953	23
EfficientNet	62.7	6	70.09	0.9102	11
MobileNetV3-Large	6.23	8.1	96.97	0.0951	68
Xception	55.1	20.53	90.91	0.2523	56

5. Conclusions

There were many advantages of deploying the new lightweight convolutional neural network model, FlexibleNet. First, we obtained the highest accuracy in qualitatively classifying Sentinel-2 images into different carbon sequestration classes. Second, the FlexibleNet model had the lowest loss values compared to the other models. Third, except for MobileNetV3-Large, the new model used the lowest number of parameters and required the lowest time. In the first experiment, the FlexibleNet model was the best one because it had the lowest number of parameters compared to the other models, including MobileNetV3-Large. In the second and third experiments, the MobileNetV3-Large model was slightly better than the FlexibleNet model, but both were stable concerning the number of parameters when the problem size changed. One disadvantage of the FlexibleNet model was its inability to overcome the MobileNetV3 model in reducing the number of parameters in all experiments. The FlexibleNet model is the first version of a series that will include enhancements to many existing features in the new model, including reducing the parameter requirements. It is also expected to be used to conduct more experiments on other complex problems, such as using tropical forest datasets.

Funding: This research received no external funding.

Data Availability Statement: All data and programs were placed on the website https://github.com/users/ma850419/FlexibleNet (accessed on 30 December 2022).

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

References

1. Fradkov, A. Early History of Machine Learning. *IFAC-Pap. OnLine* **2020**, *53*, 1385–1390. [CrossRef]
2. Wang, P.; Wang, L.; Leung, H.; Zhang, G. Super-Resolution Mapping Based on Spatial–Spectral Correlation for Spectral Imagery. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 2256–2268. [CrossRef]
3. Awad, M. Cooperative evolutionary classification algorithm for hyperspectral images. *J. Appl. Remote Sens.* **2020**, *14*, 016509. [CrossRef]
4. Masi, G.; Cozzolino, D.; Verdoliva, L.; Scarpa, G. Pansharpening by Convolutional Neural Networks. *Remote Sens.* **2016**, *8*, 594. [CrossRef]
5. 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, Columbus, OH, USA, 23–28 June 2014; pp. 580–587. [CrossRef]
6. Awad, M.M.; Lauteri, M. Self-Organizing Deep Learning (SO-UNet)—A Novel Framework to Classify Urban and Peri-Urban Forests. *Sustainability* **2021**, *13*, 5548. [CrossRef]
7. Sylvain, J.; Drolet, G.; Brown, N. Mapping dead forest cover using a deep convolutional neural network and digital aerial photography. *ISPRS J. Photogramm. Remote Sens.* **2019**, *156*, 14–26. [CrossRef]
8. Sarwinda, D.; Paradisa, R.; Bustamam, A.; Anggia, P. Deep Learning in Image Classification using Residual Network (ResNet) Variants for Detection of Colorectal Cancer. *Procedia Comput. Sci.* **2021**, *179*, 423–431. [CrossRef]
9. Tao, J.; Gu, Y.; Sun, J.; Bie, Y.; Wang, H. Research on VGG16 convolutional neural network feature classification algorithm based on Transfer Learning. In Proceedings of the 2nd China International SAR Symposium (CISS), Shanghai, China, 3–5 November 2021; pp. 1–3. [CrossRef]
10. Singh, I.; Goyal, G.; Chandel, A. AlexNet architecture based convolutional neural network for toxic comments classification. *J. King Saud Univ. Comput. Inf. Sci.* **2022**, *34*, 7547–7558. [CrossRef]

11. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9. [CrossRef]
12. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size. *arXiv* **2016**, arXiv:1602.07360. [CrossRef]
13. Chollet, F. Xception: Deep Learning with Depth wise Separable Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21—26 July 2017; pp. 1800–1807. [CrossRef]
14. Yuan, H.; Cheng, J.; Wu, Y.; Zeng, Z. Low-res MobileNet: An efficient lightweight network for low-resolution image classification in resource-constrained scenarios. *Multimed. Tools Appl.* **2022**, *81*, 38513–38530. [CrossRef]
15. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V. Searching for MobileNetV3. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Republic of Korea, 27–28 October 2019; pp. 1314–1324. [CrossRef]
16. Ma, N.; Zhang, X.; Zheng, H.T.; Sun, J. Shufflenet v2: Practical guidelines for efficient CNN architecture design. In Proceedings of the European conference on computer vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 116–131. [CrossRef]
17. Tan, M.; Le, Q.V. (2019) EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the 36th International Conference on Machine Learning, Long Beach, CA, USA, 10–19 June 2019; pp. 6105–6114.
18. Sarker, I.H. Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. *SN Comput. Sci.* **2021**, *2*, 420. [CrossRef] [PubMed]
19. Zhou, Y.; Bai, Y.; Bhattacharyya, S.; Huttunen, H. Elastic Neural Networks for Classification. In Proceedings of the 2 IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS), Taiwan, China, 18–20 March 2019; pp. 251–255. [CrossRef]
20. Bai, Y.; Bhattacharyya, S.; Happonen, A.; Huttunen, H. Elastic Neural Networks: A Scalable Framework for Embedded Computer Vision. In Proceedings of the 26th European Signal Processing Conference (EUSIPCO), Rome, Italy, 3–7 September 2018; pp. 1472–1476. [CrossRef]
21. Yu, D.; Xu, Q.; Guo, H.; Zhao, C.; Lin, Y.; Li, D. An Efficient and Lightweight Convolutional Neural Network for Remote Sensing Image Scene Classification. *Sensors* **2020**, *20*, 1999. [CrossRef] [PubMed]
22. Chen, Y.; Chen, X.; Lin, J.; Pan, R.; Cao, T.; Cai, J.; Yu, D.; Cernava, T.; Zhang, X. DFCANet: A Novel Lightweight Convolutional Neural Network Model for Corn Disease Identification. *Agriculture* **2022**, *12*, 2047. [CrossRef]
23. Kawamiya, M.; Hajima, T.; Tachiiri, K.; Watanabe, S.; Yokohata, T. Two decades of Earth system modeling with an emphasis on Model for Interdisciplinary Research on Climate (MIROC). *Prog. Earth Planet. Sci.* **2020**, *7*, 64. [CrossRef]
24. Deng, L.; Zhu, G.Y.; Tang, Z.S.; Shangguan, Z.P. Global patterns of the effects of land-use changes on soil carbon stocks. *Glob. Ecol. Conserv.* **2016**, *5*, 127–138. [CrossRef]
25. Food And Agriculture Organization of the United Nations (FAO). *Global Forest Resources Assessment 2015—How Are the World's Forests Changing?* 2nd ed.; FAO: Rome, Italy, 2016; p. 54.
26. Bernal, B.; Murray, L.T.; Pearson, T.R.H. Global carbon dioxide removal rates from forest landscape restoration activities. *Carbon Balance Manag.* **2018**, *13*, 22. [CrossRef]
27. Kim, H.; Kim, Y.H.; Kim, R.; Park, H. Reviews of forest carbon dynamics models that use empirical yield curves: CBM-CFS3, CO2FIX, CASMOFOR, EFISCEN. *For. Sci. Technol.* **2015**, *11*, 212–222. [CrossRef]
28. Liu, F.; Tan, C.; Zhang, G.; Liu, J.X. Single-wood parameters and biomass airborne LiDAR estimation of Larix olgensis. *Trans. Chin. Soc. Agric.* **2013**, *44*, 219–224.
29. Lizuka, K.; Tateishi, R. Estimation of CO_2 Sequestration by the Forests in Japan by Discriminating Precise Tree Age Category using Remote Sensing Techniques. *Remote Sens.* **2015**, *7*, 15082–15113. [CrossRef]
30. Castro-Magnani, M.; Sanchez-Azofeifa, A.; Metternicht, G.; Laakso, K. Integration of remote-sensing based metrics and econometric models to assess the socio-economic contributions of carbon sequestration in unmanaged tropical dry forests. *Environ. Sustain. Indic.* **2021**, *9*, 100100. [CrossRef]
31. Costanza, R.; de Groot, R.; Braat, L.; Kubiszewski, I.; Fioramonti, L.; Sutton, P.; Farber, S.; Grasso, M. Twenty years of ecosystem services: How far have we come and how far do we still need to go? *Ecosyst. Serv.* **2017**, *28*, 1–16. [CrossRef]
32. Hao, H.; Li, W.; Zhao, X.; Chang, Q.; Zhao, P. Estimating the Aboveground Carbon Density of Coniferous Forests by Combining Airborne LiDAR and Allometry Models at Plot Level. *Front. Plant Sci.* **2019**, *10*, 917. [CrossRef] [PubMed]
33. Kanniah, K.; Muhamad, N.; Kang, C. Remote sensing assessment of carbon storage by urban forest, IOP Conference Series: Earth and Environmental Science. In Proceedings of the 8th International Symposium of the Digital Earth (ISDE8), Kuching, Malaysia, 26–29 August 2013; Volume 18.
34. Uniyal, S.; Purohit, S.; Chaurasia, K.; Rao, S.; Amminedu, E. Quantification of carbon sequestration by urban forest using Landsat 8 OLI and machine learning algorithms in Jodhpur, India. *Sci. Direct Urban For. Urban Green.* **2022**, *67*, 127445. [CrossRef]
35. Foody, G.; Mathur, A. A relative evaluation of multiclass image classification by support vector machines. *IEEE Trans. Geosci. Remote Sens.* **2004**, *42*, 1335–1343. [CrossRef]
36. Gwal, S.; Singh, S.; Gupta, S.; Anand, S. Understanding forest biomass and net primary productivity in Himalayan ecosystem using geospatial approach. *Model. Earth Syst. Environ.* **2020**, *6*, 10. [CrossRef]

37. Kimes, D.; Nelson, R.; Manry, M.; Fung, A. Review article: Attributes of neural networks for extracting continuous vegetation variables from optical and radar measurements. *Int. J. Remote Sens.* **1998**, *19*, 2639–2663. [CrossRef]
38. Sagi, O.; Rokach, L. Approximating XGBoost with an interpretable decision tree. *Inf. Sci.* **2021**, *572*, 522–542. [CrossRef]
39. Zhang, F.; Tian, X.; Zhang, H.; Jiang, M. Estimation of Aboveground Carbon Density of Forests Using Deep Learning and Multisource Remote Sensing. *Remote Sens.* **2022**, *14*, 3022. [CrossRef]
40. Liu, W.; Wang, Z.; Liu, X.; Zeng, N.; Liu, Y.; Alsaadi, F.E. A survey of deep neural network architectures and their applications. *Neurocomputing* **2017**, *234*, 11–26. [CrossRef]
41. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
42. Li, J.; Roy, D.A. global analysis of Sentinel-2A, Sentinel-2B and Landsat-8 data revisit intervals and implications for terrestrial monitoring. *Remote Sens.* **2017**, *9*, 902. [CrossRef]
43. Lang, N.; Jetz, W.; Schindler, K.; Wegner, A. High-resolution canopy height model of the Earth. *arXiv* **2022**, arXiv:2204.08322.
44. Abdollahnejad, A.; Panagiotidis, D.; Surový, P. Forest canopy density assessment using different approaches—Review. *J. For. Sci.* **2017**, *63*, 107–116.
45. Chen, J.; Yang, S.; Li, H.; Zhang, B.; Lv, J. Research on Geographical Environment Unit Division Based on The Method of Natural Breaks (Jenks), The International Archives of the Photogrammetry. *Remote Sens. Spat. Inf. Sci.* **2013**, *3*, 47–50, 2013 ISPRS/IGU/ICA Joint Workshop on Borderlands Modelling and Understanding for Global Sustainability 2013, Beijing, China.
46. Mamoshina, P.; Vieira, A.; Putin, E.; Zhavoronkov, A. Applications of Deep Learning in Biomedicine. *Mol. Pharm.* **2016**, *13*, 1445–1454. [CrossRef]
47. Khan, A.; Sohail, A.; Zahoora, U.; Qureshi, A.S. A Survey of the Recent Architectures of Deep Convolutional Neural Networks. *Artif. Intell. Rev.* **2020**, *53*, 5455–5516. [CrossRef]
48. Xu, B.; Wang, N.; Chen, T.; Li, M. Empirical Evaluation of Rectified Activations in Convolutional Network. *arXiv* **2015**, arXiv:1505.00853.
49. Josephine, V.L.; Nirmala, A.P.; Allur, V. Impact of Hidden Dense Layers in Convolutional Neural Network to enhance Performance of Classification Model, IOP Conference Series: Materials Science and Engineering. In Proceedings of the 4th International Conference on Emerging Technologies in Computer Engineering: Data Science and Blockchain Technology (ICETCE 2021), Jaipur, India, 3–4 February 2021; Volume 1131.
50. Lee, D.; Park, C.; Tomlin, D. Effects of land-use-change scenarios on terrestrial carbon stocks in South Korea. *Landsc. Ecol. Eng.* **2015**, *11*, 47–59. [CrossRef]
51. Scott, D. Sturges' rule. *WIREs Comput. Stat.* **2009**, *1*, 303–306. [CrossRef]
52. Belavkin, R.; Pardalos, P.; Principe, J. Value of Information in the Binary Case and Confusion Matrix. *Phys. Sci. Forum* **2022**, *5*, 5008. [CrossRef]
53. Bottou, L.; Bousquet, O. The Tradeoffs of Large Scale Learning. In *Optimization for Machine Learning*; Sra, S., Nowozin, S., Stephen, J.W., Eds.; MIT Press: Cambridge, UK, 2012; pp. 351–368. ISBN 978-0-262-01646-9.
54. Asif, A.; Waris, A.; Gilani, S.; Jamil, M.; Ashraf, H.; Shafique, M.; Niazi, I.K. Performance Evaluation of Convolutional Neural Network for Hand Gesture Recognition Using EMG. *Sensors* **2020**, *20*, 1642. [CrossRef] [PubMed]
55. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520. [CrossRef]
56. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning (Adaptive Computation and Machine Learning Series)*; The MIT Press: Cambridge, MA, USA, 2016; p. 800.