

Article

# The Use of Three-Dimensional Convolutional Neural Networks to Interpret LiDAR for Forest Inventory

Elias Ayrey \* and Daniel J. Hayes 

School of Forest Resources, University of Maine, 5755 Nutting Hall, Orono, ME 04469-5755, USA;  
daniel.j.hayes@maine.edu

\* Correspondence: elias.ayrey@maine.edu; Tel.: +1-609-651-0761

Received: 6 February 2018; Accepted: 13 April 2018; Published: 23 April 2018



**Abstract:** As light detection and ranging (LiDAR) technology becomes more available, it has become common to use these datasets to generate remotely sensed forest inventories across landscapes. Traditional methods for generating these inventories employ the use of height and proportion metrics to measure LiDAR returns and relate these back to field data using predictive models. Here, we employ a three-dimensional convolutional neural network (CNN), a deep learning technique that scans the LiDAR data and automatically generates useful features for predicting forest attributes. We test the accuracy in estimating forest attributes using the three-dimensional implementations of different CNN models commonly used in the field of image recognition. Using the best performing model architecture, we compared CNN performance to models developed using traditional height metrics. The results of this comparison show that CNNs produced 12% less prediction error when estimating biomass, 6% less in estimating tree count, and 2% less when estimating the percentage of needleleaf trees. We conclude that using CNNs can be a more accurate means of interpreting LiDAR data for forest inventories compared to standard approaches.

**Keywords:** deep learning; artificial neural network; machine learning; ALS; LiDAR; enhanced forest inventory; area-based

## 1. Introduction

Over the past two decades, airborne Light Detection and Ranging (LiDAR) has become an invaluable tool for remotely quantifying the three-dimensional structure of the forest canopy. This has enabled scientists to estimate forest attributes over large areas traditionally only measurable with an intensive field campaign. Attributes predicted with LiDAR include estimates of tree count, height, species, stem volume, and aboveground biomass [1–3]. Such enhanced forest inventories (EFIs) have proven useful for research and applications in forest and wildlife ecology, forest carbon cycling, and sustainable forest management [4–6].

The typical methodology used for developing EFIs is called the *Area Based* (AB) approach. The AB approach utilizes predictive modelling to associate plot-based field measurements with explanatory variables derived from various measures of a LiDAR dataset of the same forest area. These models are then applied to make estimates of forest attributes for new areas without field measurements [7]. Numerous statistical techniques have been used to develop AB models, among which linear mixed modelling and random forest imputation are common modelling tools amongst many established modelling techniques [8,9].

Model explanatory variables derived from LiDAR data can take several forms, but mostly constitute either measurements of point height or distribution along the vertical stratum [1,2,10]. Height measurements typically constitute summary statistics that calculate the mean, maximum, and percentile heights of points within each grid cell. Distribution metrics quantify the proportion

of points found above certain height thresholds. Since their introduction in the early 2000s, these traditional height metrics (THMs) have been proven to be effective predictors for modeling forest attributes [11,12]. Common software packages currently used for LiDAR EFI modeling, including FUSION [13] and rLiDAR [14], are based on extracting THMs from point clouds.

Despite the well-established effectiveness of THMs, these metrics come with several drawbacks. Many THMs are highly correlated with one another, and so care must be taken during model development to avoid issues of multicollinearity [15]. Furthermore, THMs generated from one LiDAR data set are sometimes not stable when applied to another due to variation in acquisition parameters such as laser penetration, pulse density, and scan angle [16–18]. Finally, THMs do not quantify measures of horizontal complexity, and so the nuance of distinct tree crown shapes is lost. This may partly account for the difficulty in estimating tree count from THMs, as reported in previous studies [19–21].

Like LiDAR modelers, computer vision scientists have struggled to develop meaningful tools for quantifying remotely sensed data. Early computer vision scientists developed a series of metrics for estimating an image's content using summary statistics of color histograms, detected edges, and blobs [22]. This methodology is known as feature engineering, and bears some resemblance to LiDAR's THMs. Feature engineering, however, has largely since been discarded by the computer vision field in favor of deep learning, which can learn to self-identify important spatial features in a dataset [23].

Deep learning is a form of machine learning, and refers primarily to artificial neural networks of a sufficient complexity so as to interpret raw data without the need for human-derived explanatory variables [23]. This differs from simpler neural networks (such as perceptrons), which impute attributes using features such as THMs and have been used to estimate forest attributes in remote sensing for many years [24–26]. Convolutional neural networks (CNNs), such as those employed here, are a subdivision of deep learning and are distinct in that they interpret spatial data by scanning it using a series of trainable moving windows. The values of these windows are initially randomized. During training, the CNN uses an optimizer to tune these values to identify useful features and objects for estimating the response variable. This is accomplished without user input.

The first CNN was developed in 1995 to classify images of hand-written digits [27]. However, the technique was largely underrepresented in scientific literature for the following decade, partially owing to computational and software constraints [23]. This changed in 2012, when a deep CNN (consisting of many layered convolutions) won the ImageNet image classification competition by a wide margin [23,28]. Since then, CNNs of increasing complexity and depth have consistently outperformed models based upon feature extraction for computer vision [29–32]. Other variants, such as inception and residual models, have also been developed to extract greater numbers of meaningful features by using windows of varying sizes and preserving useful input data as it passes through the model [33].

Though CNNs have been mostly developed for computer vision with two-dimensional images, those in other fields are beginning to apply these algorithms to novel, three-dimensional problems. A few studies have begun using deep learning for measuring and analyzing forest attributes. For example, Guan et al. [34] used a segmentation technique to isolate tree crowns, and then used a neural network to classify species based on point distribution. In another study, Ghamisi et al. [35] applied a 2D CNN to estimate forest attributes from rasterized LiDAR and hyperspectral data. A 2D CNN is designed to scan two-dimensional images, and is only capable of identifying spatial features along two axes. A 3D CNN uses three-dimensional windows to scan volumetric space and identify spatial features in the X, Y, and Z axes. In this context, a 2D CNN is only capable of scanning a height map derived from a LiDAR point cloud, whereas a 3D CNN is capable of scanning the entire cloud for 3D features such as tree crowns, stems, or branches.

Presently, 3D CNNs have not been used to interpret LiDAR data for forest inventory; however, there are examples of their use in other fields. One study implemented a 3D CNN for use in airborne

LiDAR to identify helicopter landing zones in real time [36]. Others have used them in conjunction with terrestrial LiDAR to map obstacles for autonomous cars [37,38]. One common application has been to identify malignancies using 3D medical scans [39–41]. Several studies have also used 3D CNNs for household object detection [42–44].

The goal of this study is to adapt common 2D CNN implementations to scan the 3D volumetric pixels, or voxels, derived from a LiDAR point cloud to estimate aboveground biomass, tree count, and the percent of needleleaf stems. Several CNN architectures were adapted, beginning with the least complex (LeNet) [27] and working towards deeper, more contemporary architectures (Inception V3) [45]. We compare these architectures against one another, noting performance with model complexity and type. The best performing CNNs were then tested against random forest and linear mixed models trained using THM.

## 2. Materials and Methods

### 2.1. Field Data

Deep learning is renowned for requiring extremely large datasets in order to calibrate properly. Training datasets often consist of thousands, if not millions, of training examples [46]. To meet this requirement, forest inventories from eight different sites across the Northern New England/ Acadian forest region were aggregated (Table 1). Each site often contained several inventories conducted by different agencies. We note that the models we developed are not intended to represent the entire landscape, and serve only as comparisons between model types.

**Table 1.** Field inventory sites used for model training, as well as the origin of the accompanying LiDAR (see Section 2.1) and the percent abundance of the most common species (defined as the top five species composing greater than 5% of the total makeup).

Site Name	Location	Number of Plots	LiDAR Acquisition	Species *
Acadia National Park	44°20'N, 68°16'W	9	G-LIHT	69% <i>Piru</i> , 8% <i>Thoc</i> , 6% <i>Acru</i> , 5% <i>Pist</i> , 5% <i>Abba</i>
Bartlett Experimental Forest	44°2'N, 71°17'W	46	NEON	46% <i>Fagr</i> , 18% <i>Tsca</i> , 14% <i>Acru</i> , 9% <i>Acsa</i> , 5% <i>Beal</i>
University of Maine's Cooperative Forest Research Unit	45°11'N, 69°42'W	90	G-LIHT	50% <i>Piru</i> , 42% <i>Abba</i>
University of Maine's Dwight B. Demeritt Forest	44°55'N, 68°40'W	344	G-LIHT	28% <i>Acru</i> , 23% <i>Abba</i> , 10% <i>Thoc</i> , 8% <i>Tsca</i> , 7% <i>Piru</i>
Harvard Forest	42°31'N, 72°11'W	13,470	G-LIHT and NEON	32% <i>Tsca</i> , 25% <i>Acru</i> , 14% <i>Quru</i> , 6% <i>Pist</i> , 5% <i>Beal</i>
Holt Research Forest	43°52'N, 69°46'W	2002	G-LIHT	25% <i>Pist</i> , 24% <i>Acru</i> , 21% <i>Quru</i> , 7% <i>Piru</i> , 7% <i>Abba</i>
Howland Research Forest	45°12'N, 68°45'W	1014	G-LIHT	40% <i>Piru</i> , 31% <i>Tsca</i> , 11% <i>Thoc</i> , 10% <i>Acru</i>
Penobscot Experimental Forest	44°52'N, 68°37'W	301	G-LIHT	27% <i>Abba</i> , 23% <i>Tsca</i> , 13% <i>Pist</i> , 13% <i>Piru</i> , 12% <i>Acru</i>

\* Species abbreviations are as follows: *Abba* = *Abies balsamea*, *Acru* = *Acer rubrum*, *Acsa* = *Acer saccharum*, *Beal* = *Betula alleghaniensis*, *Piru* = *Picea rubens*, *Pist* = *Pinus strobus*, *Tsca* = *Tsuga canadensis*, *Thoc* = *Thuja occidentalis*, *Quru* = *Quercus rubra*.

All inventories consisted of stem-maps of varying sizes, on which all trees with diameters greater than 11.4 cm were measured. In some inventories, heights were measured on only a subset of trees, and so species-specific, non-linear height-diameter models were generated following a Chapman-Richards model form to impute tree height. Uncommon trees (primarily deciduous) were grouped with similar species; separate models were developed for the five sites requiring some height imputation. Some inventories were measured up to five years prior to the LiDAR being acquired over that site. In these cases, trees were projected forward in time using the Forest Vegetation Simulator's Acadian Variant [47]. Table 2 contains summary statistics of field inventory data. Several plots lay

entirely within canopy gaps or non-forested wetlands, and so had no trees. These were retained to better constrain the models.

**Table 2.** Summary statistics of the field inventory data used in this study.

	Min	Mean	Max	SD
Aboveground Biomass (Mg/ha)	0	185.7	715.7	88.5
Tree Count (#)	0	6.93	25	3.65
Percent Needleleaf (%)	0	42.9	100	28.9

We chose to model aboveground biomass, tree count, and the percent of needleleaf stems (including *Thuja occidentalis*), as the performance of these models may be indicative of other attributes that measure size, number, and types of trees. Biomass was estimated for each tree using the component ratio method developed for the US Forest Inventory and Analysis program [48]. This method makes use of species-specific localized stem volume and component equations, which are then converted to biomass using specific gravity and a site modifier.

The stem maps were segmented into  $10 \times 10$  m grid cells, a comparatively small cell size for this type of inventory. In using 10 m cells, we were able to amplify the amount of training data available through the use of smaller stem mapped plots while retaining an area large enough to include several whole tree crowns. In order to account for edge effects given the small plot size, regional diameter to crown width equations were used to project the area of each tree's crown in space [49]. Tree-level biomass was then multiplied by the proportion that each tree's crown overlapped the plot. Thus, trees were treated as areas containing biomass, rather than points with a binary in/out classification. All field-measured plots were larger than the 10 m grid cells used in this study, so the biomass proportion of trees with stems outside of the grid cells were also accounted for, assuring unbiased predictions across multiple cells.

One common technique for increasing sample size in deep learning model applications is to transform or rotate input images multiple times and then calibrate a model using all of these quasi-novel inputs [50,51]. In a similar vein, there were instances in which two LiDAR acquisitions overlapped a single plot, and we included both in the models. The precise configuration of LiDAR returns will always vary from one LiDAR acquisition to the next. In some instances, plots were allowed to overlap one another by a maximum of 25%. This allowed more 10 m cells to fit into many of the stem mapped areas, while retaining novel assemblages of trees. A total of 17,537 field plots were generated for training and analysis in this study.

## 2.2. LiDAR Data

Airborne LiDAR data were acquired over the study areas in a series of flights conducted by two entities. Data over 10,627 of the 17,537 field plots at seven sites were collected by NASA Goddard's LiDAR, Hyperspectral, and Thermal Imager (G-LIHT) [52]. The G-LIHT sensor suite operates a Riegl VQ-480 LiDAR sensor with a 300 kHz scan rate and a 1550 nm wavelength. The average pulse density ranged from 12–15 pls/m<sup>2</sup>. The remaining 6910 plots were flown by the National Ecological Observatory Network (NEON) using an Optech Gemini with a 167 kHz scan rate and a 1064 nm wavelength. The average pulse density was 9 pls/m<sup>2</sup>. All LiDAR data were acquired between 2012 and 2016 in leaf-on conditions between June and August.

The LiDAR data were preprocessed first by normalizing points using a digital elevation model so as to measure height above ground level. Next, the point clouds were clipped to the extent of the field plots, representing the  $10 \times 10$  m grid cells. The vertical space that we measured on each plot extended 35 m above the ground, an elevation slightly higher than the tallest tree in the field data. The point cloud was then binned into voxels measuring  $25 \times 25 \times 33$  cm; the dimensionality of the voxelized point cloud was thus  $40 \times 40 \times 106$ . We used rectangular voxels in order to reduce the dimensionality of the vertical axis and prevent horizontal features from being lost. The final dimensionality of the

voxels was determined through qualitative testing of several sizes using the most simple model form (LeNet). Each voxel was assigned the value equal to the number of points occurring within that 3D space. The input to the neural network was therefore a matrix of those dimensions, upon which the three-dimensional kernels of a CNN could be passed over.

### 2.3. Network Architecture

Convolutional neural networks consist of a series of data transformations, or layers, which can be arranged like building blocks in order to form specific network architectures. The following layer types were used in various combinations for the construction of the CNNs tested in this study:

- *Convolutional layers* are a series of moving windows passed over data with spatial or temporal relationships. The values of these windows are used as multipliers and are initially randomized, but are defined over time as the network trains. Often these are used to identify features such as edges.
- *Dropout layers* are commonly applied following a convolution. These consist of randomized removals of data. This can prevent overfitting of the network by altering architecture during the training process, preventing specific pathways from becoming relied upon [53].
- *Batch normalization* is another layer frequently applied after a convolution, and is used to standardize input data. This can speed up model training by ensuring data fall upon the same scale from which weights are initially randomly derived. These shift subtly with each training step, and so are another way of preventing model overfitting [54].
- *Activation layers* are an essential component to any neural network. They act as thresholds for data passage onto the next layer, similar to the action potential in a living neuron. The most common type of activation function is the rectified linear unit (ReLU), which is a linear gateway allowing only data with a value greater than zero to pass.
- *Pooling layers* are spatial aggregations of data. These are used primarily to reduce dimensionality and condense important information. Maximum pooling is the most common type of pooling, which takes the largest data value inside a moving window [55].

We tested five CNN architectures across a range of model complexity, assessing their performance using the coefficient of determination (pseudo- $R^2$ ), root mean squared error (RMSE), and bias. Each had to be adapted to measure 3D voxel space rather than 2D pixels. This conversion altered the dimensionality from each architecture's native format, and so we attempted to replicate each architecture's proportional changes in dimensionality using convolutions and pooling. For example, if an architecture called for a  $5 \times 5$  kernel over a  $30 \times 30$  image, we would scale our 3D window to approximately the same proportion relative to the input,  $7 \times 7 \times 18$ . This was also balanced with the dimensionality required for the following layer.

1. The first network architecture we tested was LeNet, the earliest CNN [27]. LeNet consists of two layers of convolutions each followed by maximum pooling and dropout. These feed into a further two fully connected layers (consisting of a ReLU activator and dropout) before producing a prediction.
2. The second network we tested was AlexNet [28], which was first introduced in 2012. This network is largely credited as the first 'deep' CNN, and it resulted in a dramatic increase in image classification accuracy. AlexNet consists of five layers of convolutions, each with an activator, with pooling following the first, second, and fifth convolutions. These lead into a further three fully connected layers and a final prediction.
3. Next, we implemented GoogLeNet, introduced in 2014 [29]. GoogLeNet improved upon image classification performance by introducing groups of convolutional layers called inception layers. Inception layers are bundles of convolutions of varying sizes and strides, useful for identifying different types of features. These are followed by batch normalization and activation, with the

outputs concatenated. GoogLeNet begins with three convolutional layers and pooling, and is followed by nine inception layers, with pooling after the second and seventh inception layer. The data are then funneled into a single fully connected output to achieve a prediction.

4. A series of networks succeeded GoogLeNet, including Inception-V3 [45]. This expanded upon GoogLeNet's inception layers, adding additional convolutions with four pathways by which the incoming data is analyzed before being concatenated. Inception-V3 begins with five convolution layers and two pooling layers, followed by eleven inception layers of varying compositions, and pooling layers interspersed along the way to continuously reduce dimensionality.
5. The final network architecture we tested was ResNet-50 [31]. Rather than use inception layers, ResNet-50 uses residual layers. Data entering a residual layer is subject to several convolutions, batch normalization, and activation, the results of which are added to the original input. By retaining the input values, useful information from the previous residual layers is preserved and improved upon, reducing the potential for values running through the network to drift. Our implementation of ResNet-50 contained an initial convolution, ReLU, and pooling layer, followed by sixteen residual layers (each containing three convolutions) with three pooling layers interspersed to gradually reduce dimensionality.

Each of the neural networks was implemented using Google's Tensorflow library [56]. Of the 17,537 total plots, we randomly withheld 2000. As is typical in many deep learning applications, we segmented the data into three datasets. The largest dataset was comprised of 15,537 plots and was used to train the model. A validation dataset comprised of 1000 withheld plots was used to determine the optimal stopping point of model training, which was the point at which mean squared error of these data no longer decreased. A third dataset called the testing data was comprised of another 1000 withheld plots and was used to assess model performance, as the independence of the validation dataset was compromised upon its use. In instances for which withheld plots had additional LiDAR flights over them, the additional LiDAR data were discarded.

Performance of each CNN was assessed by calculating the root mean squared error (RMSE) and bias against each independent variable of the testing dataset. In order to settle on the optimal architecture, we first trained each of the five models to estimate aboveground biomass. We selected the best performing model and then used the same architecture to train the tree count and percent needleleaf models. The assumption for this being that the architecture that best predicted biomass was also the architecture that best interpreted LiDAR features.

#### 2.4. Traditional Height Metrics Modelling

Models trained using the optimal CNN architecture were compared to models trained using both linear mixed modelling and random forest in regression mode with THMs. We withheld the same 2000 plots as before, once again separating out those for training, validation, and testing data. The THMs were generated for each plot using the rLiDAR package for the R programming language [14]. These metrics included measures of height percentiles, height summary statistics, and percentages of returns above thresholds.

For the purposes of model selection, we first generated small random forest models consisting of only 50 decision trees for each of the three forest attributes. We then generated importance values for each THM covariate using conditional variable importance, which is less prone to importance inflation due to correlated variables [57].

Using the top ten most important predictors of each attribute, we developed generalized linear mixed models with site as a random effect. We then performed reverse stepwise regression to select a final model form [58]. Percent needleleaf models were developed using a logistic form. The withheld testing data were used to assess model performance.

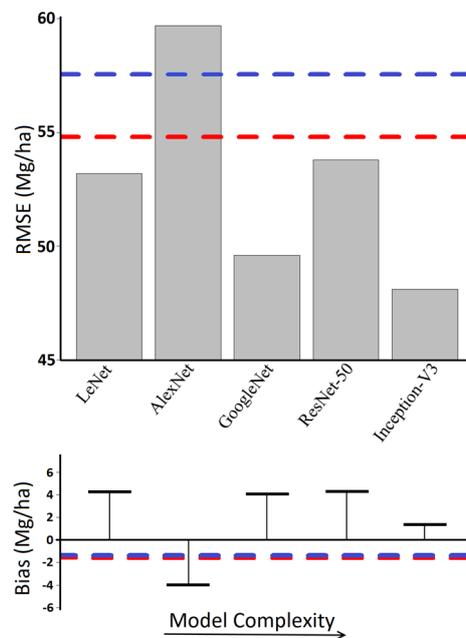
Next we developed new random forest models. Random forest can occasionally benefit from variable selection in instances in which there is a large number of unimportant variables [59]. Using the preliminary importance values, we removed 15% of the variables with the lowest importance and then

re-ran the model. We continued this process until all variables had been removed, and then used the validation data to select the model form with the lowest RMSE [59]. Fifteen percent of the THM was removed from the biomass and tree count models; none were removed from the percent needleleaf model. New models were then trained using these variables with 1000 decision trees. The withheld testing data were again used to assess model performance.

All models were graphically examined using one-to-one plots between observed field measurements and model predictions. In addition to a one-to-one line, loess regression was used to fit a moving trendline to this data. For this loess trend line, we used second order polynomials for the fit and 66% of observation to smooth the line at each value [60,61].

### 3. Results

We first compared results in terms of the RMSE and bias of aboveground biomass predictions among each of the five CNN architectures relative to the models trained using the THMs (Figure 1). Four of the five architectures we tested outperformed the both the linear mixed model with traditional height metrics (LMM-THM) and the random forest model with traditional height metrics (RF-THM) in terms of RMSE. Only the Inception-V3 CNN exhibited less absolute bias than the THM-RF model and equal absolute bias of the LMM-THM.



**Figure 1.** A comparison of biomass model performance by architecture. Results are shown in terms of root mean square error (RMSE) and bias. Red lines represent the performance of the random forest model trained on traditional height metrics. Models are listed in order of complexity, which refers to the number of trainable parameters in the model. Note that the Y-axis begins at a RMSE of 45 Mg/ha to better highlight differences in model performance.

The best performing CNN architecture was Inception-V3, which was also the most complex CNN, with the greatest number of layers and over 20 million trainable parameters. Inception-V3 had a RMSE of 48.1 Mg/ha, representing an 11% decrease in error from 54.1 Mg/ha with the RF-THM model, and a 16.5% decrease from the 57.6 Mg/ha RMSE in the LMM-THM model. The absolute bias of Inception-V3 was lower than the RF-THM and equal to the absolute bias of the LMM-THM model, at 1.3 Mg/ha. This was 68% lower than the next best performing CNN architecture, GoogleNet with 4 Mg/ha bias. The architectures using inception layers (GoogleNet, and Inception-V3) were the top two performing architectures in terms of lowest RMSE and bias.

Of the five CNN architectures, only AlexNet performed worse in terms of RMSE than the LMM-THM and RF-THM models. AlexNet's RMSE and bias were 59.7 and  $-3.9$  Mg/ha, respectively. The next-worst performing CNN was ResNet-50, with a RMSE and bias of 53.8 and 4.2 Mg/ha, respectively. Models using residual layers have garnered some degree of popularity in recent years; however, these results indicate that they may not be as effective at quantifying LiDAR, or at the very least require more fine-tuning. LeNet performed slightly better than ResNet-50 in terms of error, and equally in terms of bias despite its relative simplicity (53.3 Mg/ha and 4.2 Mg/ha).

Based on these results for modeling biomass, Inception-V3 models were assumed to be the best for interpreting LiDAR data and were developed for the other two desired forest attributes: tree count and percent of needleleaf trees. Results of those models are shown in Table 3. Each of the Inception-V3 CNNs outperformed the LMM-THM and RF-THM models in terms of RMSE; however, the improvement in accuracy was less pronounced for both the tree count and percent needleleaf models than with the biomass estimation. Random forest models consistently had a lower RMSE and higher coefficient of determination (pseudo  $R^2$ ) than the linear mixed models, while the linear mixed models had slightly lower absolute bias than random forest.

The CNN model predicting tree count resulted in 6% less error than the RF-THM model and 10% less RMSE than the LMM-THM, with a RMSE of 2.78 trees. However, the CNN model's bias was double that of THM-RF, at 0.2 and 0.1 trees, respectively. The LMM-THM exhibited almost no overall bias in estimating tree count (0.3 trees). With a  $10 \times 10$  m cell size, these biases represent between 3–20 trees/ha, which is a relatively low bias for all three models (0.5–2.8%) when taking into consideration that the mean value of these plots was 714 trees/ha.

The CNN model predicting percent needleleaf had a RMSE of 18.7%, which is 2% less than the RF-THM model's RMSE of 19.1% and 22% less than the LMM-THM model's RMSE of 24.1%. The percent needleleaf CNN also had 60% less bias than the RF-THM and 50% less absolute bias than the LMM-THM, with a bias of 0.2%. Once again it should be noted that the overall biases of all three percent needleleaf models are only several tenths of a percent, and thus are negligible in any practical context.

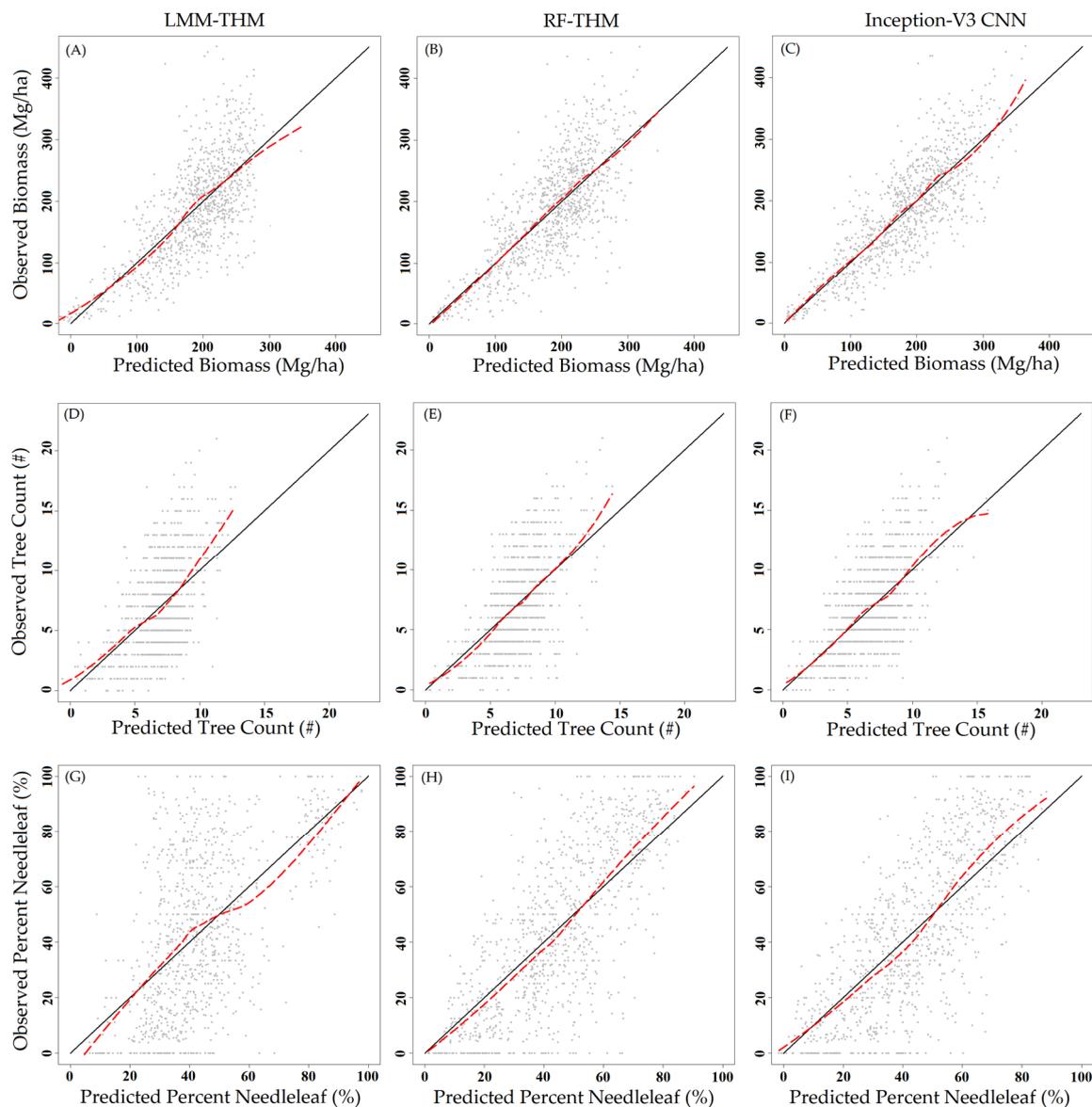
**Table 3.** Performance of each of the Inception-V3 models alongside random forest and linear mixed models (LMM) using traditional height metrics. Root Mean Square Error (RMSE) and bias were used to quantify performance. When possible, error and bias as a percent of each attribute's mean are shown in parentheses.

	Aboveground Biomass (Mg/ha)			Tree Count (#)			Percent Needleleaf (%)		
	Pseudo $R^2$	RMSE (%)	Bias (%)	Pseudo $R^2$	RMSE (%)	Bias (%)	Pseudo $R^2$	RMSE	Bias
THM with LMM	0.55	57.6 (31.3)	$-1.3$ (0.7)	0.31	3.10 (44.9)	0.03 (0.5)	0.31	24.1	$-0.4$
THM with Random Forest	0.61	54.1 (29.5)	$-1.6$ (0.9)	0.36	2.96 (43.0)	0.1 (1.4)	0.57	19.1	0.5
Inception-V3 CNN	0.69	48.1 (26.1)	1.3 (0.7)	0.43	2.78 (40.3)	0.2 (2.8)	0.58	18.7	0.2

In order to determine whether biases were consistent across plots, predicted versus observed values were plotted for each model type/forest attribute combination in Figure 2. In the estimating biomass, the LMM-THM appeared to slightly over-predict in the highest biomass plots, while the CNN appeared to slightly under-predict in the highest biomass plots (Figure 2A,C). It should be noted, however, that few plots fell within these extremes, and that loess regression is susceptible to outliers at the tail ends of data.

In predicting tree count, the LMM-THM and RF-THM both appeared to underestimate the number of trees in plots with high tree counts (Figure 2D,E). This trend appears to be less substantial in the CNN model, despite having slightly more overall bias than both the THM based models. In predicting percent needleleaf, the LMM-THM appeared to over-predict in plots with more needleleaf stems (Figure 2G). Error was also greatest for this model around plots with a mix of species. The RF-THM and the CNN models both tended to slightly over-predict percent needleleaf in plots with few needleleaf

trees, and under-predict in plots with a higher percentage of needleleaf trees. Overall, however, there appeared to be no obvious trend in observed vs. fitted biases with model type.



**Figure 2.** Predicted vs. observed plots for each of the three model types estimating biomass, tree count, and percent needleleaf. The solid line is a one-to-one line, and the dashed red line is a loess regression fit of the data. **Left (A, D, and G):** linear mixed models with traditional height metrics (LMM-THM). **Center (B, E, and H):** random forest models with traditional height metrics (RF-THM). **Right (C, F, and I):** Inception-V3 convolutional neural networks (Inception-V3 CNN).

#### 4. Discussion

Our results indicate that 3D CNNs can be used to develop an area-based forest inventory with less error and often less bias than a model based upon traditional height metrics. Model performance varied based on the specifics of the CNN architecture: those CNNs that made use of inception layers (GoogleNet and Inception-V3) outperformed those that did not. We also note that the deeper inception-based CNN, Inception-V3, outperformed the shallower inception-based GoogleNet. The CNN employing residual layers (ResNet-50) performed relatively poorly. We also evaluated ResNet-35 and ResNet 101 models, and obtained poorer and similar results, respectively. However, the

top-performing CNNs for image recognition presently make use of layers that combine residual and inception elements [45]; it is possible that such a model may outperform those tested here.

In general, the RF-THM models outperformed the LMM-THM models. Random forest models resulted in lower RMSE and greater explained variance in terms of pseudo- $R^2$ . The linear mixed models often had slightly lower absolute bias than the random forest models, but this benefit was at times negated by greater bias in plots representing extremes. These findings match those of others who have likewise concluded that random forest performs equal to or better than linear modelling for LiDAR inventories [8,60]. It should also be noted that although the THM we tested here are the most popular means of measuring LiDAR for forest attribute estimation, other studies have extracted different features from LiDAR and accompanying spectral data that may perform differently [62–64].

Performance of the CNNs relative to the THM-based models varied by the forest attribute being predicted, though in every instance the CNNs produced a lower RMSE. In modelling biomass and percent needleleaf, the CNNs performed equal to or better than THM-based models in terms of bias. The tree count CNN resulted in slightly more bias than both the THM models; however, in practical terms this bias was minimal, and both the THM based models appeared to underestimate tree count to a greater extent in plots with greater numbers of trees.

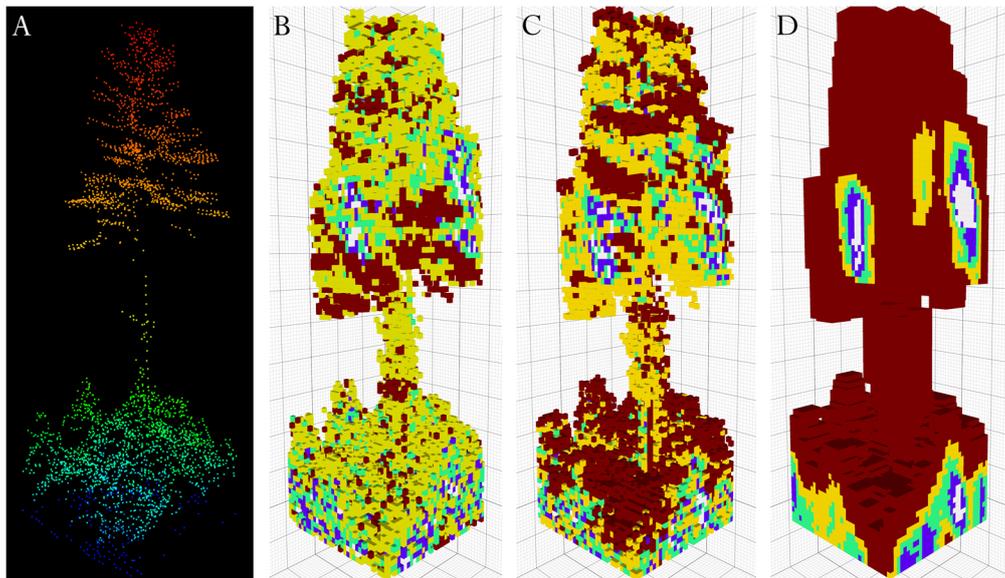
We note that the greatest performance gains of using CNNs over the THM-based models in terms of RMSE and explained variance were achieved when estimating biomass. However, it should also be noted that we also spent a greater amount of time and effort modelling biomass, as it was the attribute used to decide upon a model architecture. Neural networks offer modelers a great number of user-specified hyperparameters and architecture decisions. We believe that model performance was at least somewhat related to the amount of time spent manually fine-tuning these during training. This usually amounts to more-or-less a game of trial and error, made harder given the lengthy time it takes to train these models. Some studies have automated this process through random searches or more sophisticated means of optimization [65].

Many in the literature have noted that CNNs require a very large quantity of training data to achieve optimal performance [28,66]. This will likely relegate the use of CNNs to only those instances in which multiple forest inventories are combined, as was the case here, or when a large national forest inventory dataset is used. We did, however, demonstrate here that the number of plots used to train these models can be artificially inflated through the collection of coincident remote sensing data. This mirrors results found by other studies [67,68], in which dataset size was increased via transformations and the inclusion of many images rapidly taken of the same object. Several studies have also achieved good results by generating artificial training data [69,70], and we believe that it may be possible to do the same with LiDAR. For example, artificial plots could be generated by combining pieces of other plots, individual tree crowns, or crowns derived from allometry or forest modeling [71].

In terms of computational performance, we made no formal effort to assess the time it took to train each model included in this study. We did note, however, that training the more complex models (such as Inception V3 and Resnet-50) could take one or more days using GPU acceleration with an Nvidia Tesla k80. As settling upon the optimal model architecture and hyperparameters will likely take many attempts, the modelling process may take days or weeks. A process called “warm starting” may offer future modelers a potential solution by allowing them to reuse some or all parameters of a pre-trained 3D CNN (including those developed here), reducing the amount of training time and field plots required [72], see Supplementary Materials. We would also note that once the models were trained, predictions could be made very rapidly at a landscape level, perhaps offsetting the initial cost in training time.

Critics of deep learning have justifiably noted that the results of a CNN may be more difficult to explain given that these models lack human intuited covariates. This, combined with the size of the models, makes it nearly impossible to trace the route of data through these models and justify the result. It is possible however to visualize features identified by early layers of the model to get an understanding of the shapes and patterns being used by the model to make predictions.

These visualizations are often referred to as feature maps [73,74]. The raw values of a feature map have little biological relevance, although the relative values score features according to their utility in model predictions. The higher relative values on a feature map highlight features that are more likely to be retained by the model following an activation function. Figure 3 illustrates this with example feature maps generated over a plot. We note that some convolution layers appear to be identifying edges (Figure 3D), while others appear to be identifying surfaces and possibly branches (Figure 3B,C).



**Figure 3.** (A) The initial point cloud prior to being input to the CNN model, colored by height; (B–D) feature maps resulting from the first layer of convolutions using LeNet. Red values represent areas of higher interest to the model, blue and white represent areas of lower interest. Note that each convolution detects different patterns and features in the voxelized point cloud.

In addition to area-based forest inventories, we believe that CNNs may also be able to address the issue of individual tree segmentation. A wide array of algorithms has been put forward to segment the crowns of individual trees from a LiDAR point cloud for the purpose of developing a tree-list inventory [75–77]. Concurrently, CNNs have been enormously effective at segmenting objects from photographic and video imagery [78,79]. Most CNN-based segmentation algorithms work by identifying potential bounding boxes of objects and then analyzing the interior of those bounding boxes to assess their validity. We believe that a similar algorithm could be adapted to identify the 3D bounding boxes of individual trees. Another CNN-based segmentation method known as semantic segmentation seeks to isolate individual pixels (or voxels) that represent a desired object [80]. We believe that this method could be of use in classifying objects or terrain in LiDAR point clouds.

Another intriguing potential use for CNNs is in the development of pseudo-LiDAR point clouds. A number of studies have demonstrated the ability of CNNs to essentially be trained in reverse, producing images from other images or data [81,82]. These types of CNNs are known as deconvolutional, inverse graphic, or transposed neural networks. For example, a CNN could be trained to produce voxelized point clouds over a forested area from either previous LiDAR or a standard forest inventory. These could then be used to inform modelers as to which features their neural networks are making use of, to test ecological hypotheses, to aid in visualization, and perhaps even to project LiDAR point clouds forward in time.

We have demonstrated that deep-learning methods using CNNs to interpret LiDAR data sets can improve upon traditional methods for area-based predictions of forest attributes. However, these

improvements come with some drawbacks. Large amounts of training data, time, and effort are required for any modeling application that uses deep learning. Ultimately, it falls upon modelers to use their own best judgment to decide whether improvements in model performance are worth the effort involved in successfully training these models. That said, given our success and the prevalent adoption of deep learning in related fields, it is safe to assume that deep learning will play a large role in remote sensing in the future.

## 5. Conclusions

We demonstrate here that deep learning via 3D convolutional neural networks can be trained to interpret LiDAR point clouds for the estimation of forest attributes such as biomass, tree count, and the percent of needleleaf trees. Our results indicate that CNNs of greater complexity and those that make use of inception layers are most effective at prediction. Of the models tested, the optimal model form was a 3D variant of Google's Inception-V3. Given enough training data and fine tuning, these models frequently outperformed traditional feature-based approaches for modeling LiDAR.

**Supplementary Materials:** Additional material including relevant computer code and pre-trained models can be found online at <https://github.com/Eayrey/3D-Convolutional-Neural-Networks-with-LiDAR>.

**Acknowledgments:** We thank Bruce Cook and NASA Goddard's G-LiHT team, as well as the National Ecological Observatory Network for the use of their LiDAR. We thank Kathryn Miller and the National Park Service, David Orwig and Harvard Forest, Shawn Fraver and Howland Research Forest, and Laura Kenefic and the U.S. Forest Service's Penobscot Experimental Forest for the use of their field inventory data. We also thank Jennifer G. Fortier for her extensive revisions, John B. Kilbride for conceptualization, and Stephen Cousins with the University of Maine's Advanced Computing Group for his assistance with cluster computing.

Maine Agricultural and Forest Experiment Station Publication Number 3596. This project was supported by the USDA National Institute of Food and Agriculture, McIntire-Stennis project number #ME0-41907 through the Maine Agricultural & Forest Experiment Station.

**Author Contributions:** Elias Ayrey designed this study, developed the methodology, performed the analysis, and wrote the manuscript. Daniel J. Hayes provided extensive revisions, aided in conceptualization, and supplied laboratory and computing resources.

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

## References

1. Lim, K.S.; Treitz, P.M. Estimation of above ground forest biomass from airborne discrete return laser scanner data using canopy-based quantile estimators. *Scand. J. For. Res.* **2004**, *19*, 558–570. [[CrossRef](#)]
2. Næsset, E. Predicting forest stand characteristics with airborne scanning laser using a practical two-stage procedure and field data. *Remote Sens. Environ.* **2002**, *80*, 88–99. [[CrossRef](#)]
3. White, J.C.; Coops, N.C.; Wulder, M.A.; Vastaranta, M.; Hilker, T.; Tompalski, P. Remote sensing technologies for enhancing forest inventories: A review. *Can. J. Remote Sens.* **2016**, *42*, 619–641. [[CrossRef](#)]
4. Hudak, A.T.; Strand, E.K.; Vierling, L.A.; Byrne, J.C.; Eitel, J.U.; Martinuzzi, S.; Falkowski, M.J. Quantifying aboveground forest carbon pools and fluxes from repeat LiDAR surveys. *Remote Sens. Environ.* **2012**, *123*, 25–40. [[CrossRef](#)]
5. Graf, R.F.; Mathys, L.; Bollmann, K. Habitat assessment for forest dwelling species using LiDAR remote sensing: Capercaillie in the Alps. *For. Ecol. Manag.* **2009**, *257*, 160–167. [[CrossRef](#)]
6. Woods, M.; Pitt, D.; Penner, M.; Lim, K.; Nesbitt, D.; Etheridge, D.; Treitz, P. Operational implementation of a LiDAR inventory in Boreal Ontario. *For. Chron.* **2011**, *87*, 512–528. [[CrossRef](#)]
7. White, J.C.; Wulder, M.A.; Varhola, A.; Vastaranta, M.; Coops, N.C.; Cook, B.D.; Pitt, D.; Woods, M. A best practices guide for generating forest inventory attributes from airborne laser scanning data using an area-based approach. *For. Chron.* **2013**, *89*, 722–723. [[CrossRef](#)]
8. Penner, M.; Pitt, D.G.; Woods, M.E. Parametric vs. nonparametric LiDAR models for operational forest inventory in boreal Ontario. *Can. J. Remote Sens.* **2013**, *39*, 426–443.
9. Latifi, H.; Nothdurft, A.; Koch, B. Non-parametric prediction and mapping of standing timber volume and biomass in a temperate forest: Application of multiple optical/LiDAR-derived predictors. *Forestry* **2010**, *83*, 395–407. [[CrossRef](#)]

10. Means, J.E.; Acker, S.A.; Fitt, B.J.; Renslow, M.; Emerson, L.; Hendrix, C.J. Predicting forest stand characteristics with airborne scanning lidar. *Photogramm. Eng. Remote Sens.* **2000**, *66*, 1367–1372.
11. Nilsson, M.; Nordkvist, K.; Jonzén, J.; Lindgren, N.; Axensten, P.; Wallerman, J.; Egberth, M.; Larsson, S.; Nilsson, L.; Eriksson, J. A nationwide forest attribute map of Sweden predicted using airborne laser scanning data and field data from the National Forest Inventory. *Remote Sens. Environ.* **2017**, *194*, 447–454. [[CrossRef](#)]
12. Goodbody, T.R.; Coops, N.C.; Tompalski, P.; Crawford, P.; Day, K.J. Updating residual stem volume estimates using ALS-and UAV-acquired stereo-photogrammetric point clouds. *Int. J. Remote Sens.* **2017**, *38*, 2938–2953. [[CrossRef](#)]
13. McGaughey, R.J. *FUSION/LDV: Software for LIDAR Data Analysis and Visualization*; US Department of Agriculture, Forest Service, Pacific Northwest Research Station: Seattle, WA, USA, 2009; Volume 123.
14. Silva, C.A.; Crookston, N.L.; Hudak, A.T.; Vierling, L.A. rLiDAR: An R Package for Reading, Processing and Visualizing LiDAR (Light Detection and Ranging) Data, Version 0.1. 2015. Available online: <https://cran.r-project.org/package=rLiDAR> (accessed on 12 December 2017).
15. Junttila, V.; Kauranne, T.; Finley, A.O.; Bradford, J.B. Linear models for airborne-laser-scanning-based operational forest inventory with small field sample size and highly correlated LiDAR data. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 5600–5612. [[CrossRef](#)]
16. Goodwin, N.R.; Coops, N.C.; Culvenor, D.S. Assessment of forest structure with airborne LiDAR and the effects of platform altitude. *Remote Sens. Environ.* **2006**, *103*, 140–152. [[CrossRef](#)]
17. Roussel, J.R.; Caspersen, J.; Béland, M.; Thomas, S.; Achim, A. Removing bias from LiDAR-based estimates of canopy height: Accounting for the effects of pulse density and footprint size. *Remote Sens. Environ.* **2017**, *198*, 1–16. [[CrossRef](#)]
18. Holmgren, J.; Nilsson, M.; Olsson, H. Simulating the effects of lidar scanning angle for estimation of mean tree height and canopy closure. *Can. J. Remote Sens.* **2003**, *29*, 623–632. [[CrossRef](#)]
19. Treitz, P.; Lim, K.; Woods, M.; Pitt, D.; Nesbitt, D.; Etheridge, D. LiDAR sampling density for forest resource inventories in Ontario, Canada. *Remote Sens.* **2012**, *4*, 830–848. [[CrossRef](#)]
20. Hayashi, R.; Weiskittel, A.; Sader, S. Assessing the feasibility of low-density LiDAR for stand inventory attribute predictions in complex and managed forests of northern Maine, USA. *Forests* **2014**, *5*, 363–383. [[CrossRef](#)]
21. Shang, C.; Treitz, P.; Caspersen, J.; Jones, T. Estimating stem diameter distributions in a management context for a tolerant hardwood forest using ALS height and intensity data. *Can. J. Remote Sens.* **2017**, *43*, 79–94. [[CrossRef](#)]
22. Antani, S.; Kasturi, R.; Jain, R. A survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video. *Pattern Recognit.* **2002**, *35*, 945–965. [[CrossRef](#)]
23. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436. [[CrossRef](#)] [[PubMed](#)]
24. Jin, Y.Q.; Liu, C. Biomass retrieval from high-dimensional active/passive remote sensing data by using artificial neural networks. *Int. J. Remote Sens.* **1997**, *18*, 971–979. [[CrossRef](#)]
25. Joibary, S.S. Forest attributes estimation using aerial laser scanner and TM data. *For. Syst.* **2013**, *22*, 484–496.
26. Niska, H.; Skon, J.P.; Packalen, P.; Tokola, T.; Maltamo, M.; Kolehmainen, M. Neural networks for the prediction of species-specific plot volumes using airborne laser scanning and aerial photographs. *IEEE Trans. Geosci. Remote Sens.* **2010**, *48*, 1076–1085. [[CrossRef](#)]
27. LeCun, Y.; Bengio, Y. Convolutional networks for images, speech, and time series. *Handb. Brain Theory Neural Netw.* **1995**, *3361*, 255–258.
28. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*; MIT Press Ltd.: Cambridge, MA, USA, 2012; pp. 1097–1105.
29. 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 Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 8–10 June 2015.
30. Taigman, Y.; Yang, M.; Ranzato, M.; Wolf, L. Deepface: Closing the gap to human-level performance in face verification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1701–1708.
31. 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, NV, USA, 27–30 June 2016; pp. 770–778.

32. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the AAAI-17: Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; Volume 4, p. 12.
33. Smith, L.N.; Topin, N. Deep convolutional neural network design patterns. *arXiv* **2016**, arXiv:1611.00847.
34. Guan, H.; Yu, Y.; Ji, Z.; Li, J.; Zhang, Q. Deep learning-based tree classification using mobile LiDAR data. *Remote Sens. Lett.* **2015**, *6*, 864–873. [[CrossRef](#)]
35. Ghamisi, P.; Höfle, B.; Zhu, X.X. Hyperspectral and LiDAR data fusion using extinction profiles and deep convolutional neural network. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2017**, *10*, 3011–3024. [[CrossRef](#)]
36. Maturana, D.; Scherer, S. 3D convolutional neural networks for landing zone detection from lidar. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 3471–3478.
37. Li, B. 3D fully convolutional network for vehicle detection in point cloud. *arXiv* **2016**, arXiv:1611.08069.
38. Matti, D.; Ekenel, H.K.; Thiran, J. Combining LiDAR space clustering and convolutional neural networks for pedestrian detection. *arXiv* **2017**, arXiv:1710.06160.
39. Prasoon, A.; Petersen, K.; Igel, C.; Lauze, F.; Dam, E.; Nielsen, M. Deep feature learning for knee cartilage segmentation using a triplanar convolutional neural network. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 246–253.
40. Kamnitsas, K.; Ledig, C.; Newcombe, V.F.; Simpson, J.P.; Kane, A.D.; Menon, D.K.; Rueckert, D.; Glocker, B. Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. *Med. Image Anal.* **2017**, *36*, 61–78. [[CrossRef](#)] [[PubMed](#)]
41. Yi, D.; Zhou, M.; Chen, Z.; Gevaert, O. 3-D Convolutional Neural Networks for Glioblastoma Segmentation. *arXiv* **2016**, ArXiv:1611.04534.
42. Maturana, D.; Scherer, S. Voxnet: A 3d convolutional neural network for real-time object recognition. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 922–928.
43. Qi, C.R.; Su, H.; Nießner, M.; Dai, A.; Yan, M.; Guibas, L.J. Volumetric and multi-view CNNs for object classification on 3d data. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 5648–5656.
44. Wu, J.; Zhang, C.; Xue, T.; Freeman, B.; Tenenbaum, J. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in Neural Information Processing Systems*; MIT Press Ltd.: Cambridge, MA, USA, 2016; pp. 82–90.
45. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 2818–2826.
46. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [[CrossRef](#)]
47. Weiskittel, A.; Russell, M.; Wagner, R.; Seymour, R. *Refinement of the Forest Vegetation Simulator Northeast Variant Growth and Yield Model: Phase III*; Cooperative Forestry Research Unit—University of Maine: Orono, ME, USA, 2012; pp. 96–104.
48. Woodall, C.W.; Heath, L.S.; Domke, G.M.; Nichols, M.C. *Methods and Equations for Estimating Aboveground Volume, Biomass, and Carbon for Trees in the US Forest Inventory, 2010*; United States Department of Agriculture: Washington, DC, USA, 2011.
49. Russell, M.B.; Weiskittel, A.R. Maximum and largest crown width equations for 15 tree species in Maine. *North J. Appl. For.* **2011**, *28*, 84–91.
50. Amaral, T.; Silva, L.M.; Alexandre, L.A.; Kandaswamy, C.; de Sá, J.M.; Santos, J.M. Transfer learning using rotated image data to improve deep neural network performance. In *International Conference Image Analysis and Recognition*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 290–300.
51. Paulin, M.; Revaud, J.; Harchaoui, Z.; Perronnin, F.; Schmid, C. Transformation pursuit for image classification. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 24–27 June 2014; pp. 3646–3653.

52. Cook, B.D.; Nelson, R.F.; Middleton, E.M.; Morton, D.C.; McCorkel, J.T.; Masek, J.G.; Ranson, K.J.; Ly, V.; Montesano, P.M. NASA Goddard's LiDAR, hyperspectral and thermal (G-LiHT) airborne imager. *Remote Sens.* **2013**, *5*, 4045–4066. [[CrossRef](#)]
53. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
54. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 448–456.
55. Hinton, G.E.; Srivastava, N.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R.R. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv* **2012**, ArXiv:1207.0580.
56. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv* **2016**, ArXiv:1603.04467.
57. Strobl, C.; Boulesteix, A.-L.; Kneib, T.; Augustin, T.; Zeileis, A. Conditional variable importance for random forests. *BMC Bioinform.* **2008**, *9*, 307. [[CrossRef](#)] [[PubMed](#)]
58. Kuznetsova, A.; Brockhoff, P.B.; Christensen, R.H.B. Package 'lmerTest'. R Package Version 2.0. 2015. Available online: <https://cran.r-project.org/package=lmerTest> (accessed on 12 December 2017).
59. Diaz-Uriarte, R.; de Andrés, S.A. Variable selection from random forests: Application to gene expression data. *arXiv* **2005**, arXiv:q-bio/0503025.
60. Hayashi, R.; Weiskittel, A.; Kershaw, J.A., Jr. Influence of prediction cell size on LiDAR-derived area-based estimates of total volume in mixed-species and multicohort forests in northeastern North America. *Can. J. Remote Sens.* **2016**, *42*, 473–488. [[CrossRef](#)]
61. Cleveland, W.S. Robust locally weighted regression and smoothing scatterplots. *J. Am. Stat. Assoc.* **1979**, *74*, 829–836. [[CrossRef](#)]
62. Bouvier, M.; Durrieu, S.; Fournier, R.A.; Renaud, J.P. Generalizing predictive models of forest inventory attributes using an area-based approach with airborne LiDAR data. *Remote Sens. Environ.* **2015**, *156*, 322–334. [[CrossRef](#)]
63. Tuominen, S.; Haapanen, R. Estimation of forest biomass by means of genetic algorithm-based optimization of airborne laser scanning and digital aerial photograph features. *Silva Fenn.* **2013**, *47*. [[CrossRef](#)]
64. Véga, C.; Renaud, J.P.; Durrieu, S.; Bouvier, M. On the interest of penetration depth, canopy area and volume metrics to improve Lidar-based models of forest parameters. *Remote Sens. Environ.* **2016**, *175*, 32–42. [[CrossRef](#)]
65. Bergstra, J.; Bengio, Y. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* **2012**, *13*, 281–305.
66. Pinto, N.; Cox, D.D.; DiCarlo, J.J. Why is real-world visual object recognition hard? *PLoS Comput. Biol.* **2008**, *4*, e27. [[CrossRef](#)] [[PubMed](#)]
67. Taylor, L.; Nitschke, G. Improving Deep Learning using Generic Data Augmentation. *arXiv* **2017**, ArXiv:1708.06020.
68. Simonyan, K.; Zisserman, A. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*; MIT Press Ltd.: Cambridge, MA, USA, 2014; pp. 568–576.
69. Peng, X.; Sun, B.; Ali, K.; Saenko, K. Learning deep object detectors from 3D models. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1278–1286.
70. Ros, G.; Sellart, L.; Materzynska, J.; Vazquez, D.; Lopez, A.M. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 3234–3243.
71. Fischer, R.; Bohn, F.; de Paula, M.D.; Dislich, C.; Groeneveld, J.; Gutiérrez, A.G.; Kazmierczak, M.; Knapp, N.; Lehmann, S.; Paulick, S. Lessons learned from applying a forest gap model to understand ecosystem and carbon dynamics of complex tropical forests. *Ecol. Model.* **2016**, *326*, 124–133. [[CrossRef](#)]
72. Shin, H.-C.; Roth, H.R.; Gao, M.; Lu, L.; Xu, Z.; Nogues, I.; Yao, J.; Mollura, D.; Summers, R.M. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE Trans. Med. Imaging* **2016**, *35*, 1285–1298. [[CrossRef](#)] [[PubMed](#)]
73. Zeiler, M.D.; Fergus, R. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 818–833.

74. Yosinski, J.; Clune, J.; Nguyen, A.; Fuchs, T.; Lipson, H. Understanding neural networks through deep visualization. *arXiv* **2015**, ArXiv:1506.06579.
75. Hyypä, J.; Hyypä, H.; Leckie, D.; Gougeon, F.; Yu, X.; Maltamo, M. Review of methods of small footprint airborne laser scanning for extracting forest inventory data in boreal forests. *Int. J. Remote Sens.* **2008**, *29*, 1339–1366. [[CrossRef](#)]
76. Popescu, S.C. Estimating biomass of individual pine trees using airborne lidar. *Biomass Bioenergy* **2007**, *31*, 646–655. [[CrossRef](#)]
77. Ayrey, E.; Fraver, S.; Kershaw, J.A., Jr.; Kenefic, L.S.; Hayes, D.; Weiskittel, A.R.; Roth, B.E. Layer Stacking: A Novel Algorithm for Individual Forest Tree Segmentation from LiDAR Point Clouds. *Can. J. Remote Sens.* **2017**, *43*, 16–27. [[CrossRef](#)]
78. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 779–788.
79. Szegedy, C.; Toshev, A.; Erhan, D. Deep neural networks for object detection. In *Advances in Neural Information Processing Systems*; MIT Press Ltd.: Cambridge, MA, USA, 2013; pp. 2553–2561.
80. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
81. Kulkarni, T.D.; Whitney, W.F.; Kohli, P.; Tenenbaum, J. Deep convolutional inverse graphics network. In *Advances in Neural Information Processing Systems*; MIT Press Ltd.: Cambridge, MA, USA, 2015; pp. 2539–2547.
82. Yan, X.; Yang, J.; Sohn, K.; Lee, H. Attribute2image: Conditional image generation from visual attributes. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 776–791.



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).