# DEEP MULTI-TASK LEARNING FOR TREE GENERA CLASSIFICATION

Connie Ko[1*], Jungwon Kang[1], Gunho Sohn[1]

[1]Department of Earth and Space Science and Engineering, Lassonde School of Engineering, York University, Toronto, Canada (cko*, jkang99, gsohn)@yorku.ca

**Commission II, WGII/3**

**KEY WORDS**: LiDAR, multi-task learning, multiple-view data generation, deep neural network, tree classification

## ABSTRACT:

The goal for our paper is to classify tree genera using airborne Light Detection and Ranging (LiDAR) data with Convolution Neural Network (CNN) - Multi-task Network (MTN) implementation. Unlike Single-task Network (STN) where only one task is assigned to the learning outcome, MTN is a deep learning architect for learning a main task (classification of tree genera) with other tasks (in our study, classification of coniferous and deciduous) simultaneously, with shared classification features. The main contribution of this paper is to improve classification accuracy from CNN-STN to CNN-MTN. This is achieved by introducing a concurrence loss ($L_{\text{cd}}$) to the designed MTN. This term regulates the overall network performance by minimizing the inconsistencies between the two tasks. Results show that we can increase the classification accuracy from 88.7% to 91.0% (from STN to MTN). The second goal of this paper is to solve the problem of small training sample size by multiple-view data generation. The motivation of this goal is to address one of the most common problems in implementing deep learning architecture, the insufficient number of training data. We address this problem by simulating training dataset with multiple-view approach. The promising results from this paper are providing a basis for classifying a larger number of dataset and number of classes in the future.

## 1. INTRODUCTION

The use of airborne LiDAR for tree species classification has proven its effectiveness with high success rate. Compared to traditional aerial imageries, a new set of crown structural variables being able to derive from the three-dimension (3D) data, a generation of studies has evolved based on this technology. Numerous studies have been conducted to use LiDAR only (e.g. Lim et al., 2003; Holmgren and Persson 2004; Brandtbert, 2007; Yao et al., 2012) and a combination of LiDAR and spectral signature for tree species classification (e.g. Holmgren et al., 2008; Jones et al., 2010, Liu et al., 2011, Alonzo et al., 2014) with promising accuracy. Internal tree crown geometry metrics such as branching structures and external tree crown geometry metrics such as overall shape of the tree crown has also been taken into consideration in the design of classification features (Ko et al., 2013; Blomley et al., 2017). With the increasing computer power, lower sensor cost and more complex algorithm, LiDAR data are being more readily available, and the field of research had grown significantly in the recent years. However, these researches rely on the derivation of hand-crafted classification features, which needs a lot of human intervention. Also, hand-crafted classification features are sensitive to local changes, a classification that works well in a study area may not work well in another.

In this paper, we propose a unique and efficient approach of classifying tree genera from LiDAR collected data by a deep learning approach. One of the advantages is the needlessness to design hand-crafted classification features. This way, when the number of classes increases in the future, there is no need to redesign the classification features, even with the change in geographical location. Often the same tree genera (or species) growing in different environment may appear differently and may require additional classification features when new datasets are added. Also, trees that are different in ages may appear differently; often these are the challenges in tree genera classification. A universal set of classification features for all tree genera (species) is almost impossible. However with deep learning, classification features can be learnt through the representation of the dataset. Also, rather than classifying tree genera as a single objective (or single task), we assigned two tasks for a classification objective (coniferous-deciduous and genera classification). Intuitively, same tree genera share the same classification features, but we also observe that tree belong to the same group, coniferous or deciduous also share common classification features within the group. We propose a new way of classifying tree genera that has two unique advancements. The first is the use of Multi-task Network and the second is the introduction of a constraint term (concurrence loss ($L_{\text{cd}}$)) between the tasks for improving overall accuracies.

### 1.1 Deep Convolution Networks

The field of classification (and object recognition) has a very long history, this field falls into a wide discipline of Artificial Intelligence (AI) (Goodfellow et al., 2016) and has been applied to many domains of science such as speech recognition, computer vision and many more (LeCun et al., 2015). In environmental science, the use of remotely sensed imagery for classifying natural objects falls into this AI category and there exist a wide range of literature that discuss the methodologies for supervised learning with various machine learning methods.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-IV-2-153-2018

Until recently, a subset of machine learning methods, deep learning has been actively researched and applied in different fields. The application of deep learning has attracted much attention since it has dramatically improved the classification results with outstanding classification accuracy (LeCun et al., 2015). Deep learning falls into the category of representation learning where classification features are derived from the representation of the dataset itself (Goodfellow et al., 2016, Chapter 15). As opposed to classical feature learning where classification features are hand-crafted and designed depending on the nature of the raw dataset. This approach often requires some prior knowledge of the raw data as well. Few benchmarking examples of deep learning research include LeNet-5 (LeCun et al., 1998), ImageNet (Dent et al., 2009), GoogleLeNet (Szegedy et al., 2015) and Microsoft ResNet (He et al., 2016). The needlessness of feature derivation is the major advantage of representation learning over feature learning; it is a more automatic approach where there is less prior knowledge needed for the dataset.

Some of the earlier work of the deep learning model originated from artificial neural networks (ANNs) and was analogical to the neural networks of the brain. Where learning is breaking down into layers (input layer, hidden layers and output layer) and the learning model is an assembly of inter-connected nodes (called neurons) and weighted links. The term “deep” from deep learning comes from neural network composed with multiple hidden layers. Convolutional Neural Network (CNNs) is one of the most popular feed forward deep learning architectures because of its high performance accuracy. As a result, we adopted a CNN architecture for our tasks. Although CNN can be implemented in 3D space, we limit our experiments in 2D space for this study and the 3D convolution will be addressed in the future study. The use of CNNs adapted to 3D data is not discussed in this paper but several approaches can be found in examples such as Boulch et al. (2017), Huang and You (2016) and Lawin et al. (2017).

### 1.2 Multiple-view Data Generation

One of the most challenging problems in deep learning (or any learning problems) is the limited amount of trained dataset. A large pre-labeled dataset set is required to train the network for learning and often, the acquisition of these pre-labeled data is costly. Data augmentation refers to the generation of new dataset by providing transformations (e.g. scaling, rotating, and translating) to the original data. Its aim is to increase the number of training data and it has been proven the effectiveness by avoiding a well-known overfitting problem in image classification (Krizhevsky et al., 2012). Data augmentation also addresses the problem of imbalance training and can improve CNN-based methods (Chatfield et al., 2014). In 3D, a particular way of data augmentation is to generate a set of 2D images from viewing an object at different angles, called multi-view data generation (Su et al., 2015). We transformed the 3D LiDAR trees into 2D image space for using the existing tools that are already developed for the deep-learning algorithms.

### 1.3 Multi-task Network (MTN) for Tree Genera

MTN has been applied in different areas such as computer vision and drug discovery with success (Ramsundar et al., 2015; Girshick, 2015). As the name suggests, MTN learn more than one task at the same time with shared classification features. In our study, the two tasks are coniferous-deciduous classification and genera classification.

In general, by performing multiple tasks simultaneously, classification features derived from the classifier are less likely to be tailored for one single task and therefore could perform better by avoiding overfitting the classification problem. According to Ruder (2017), MTN is successful because learning from different tasks average noise patterns observed in different tasks. Also, in the case where data is very noisy, features derived from the first task can be verified with its relevancy by other tasks. Moreover, the network can learn certain features from a task that is easier for the feature to learn from and by introducing an inductive bias. Thus, MTN can reduces the risk of overfitting, resulting better classification accuracy.

## 2. STUDY AREA AND MATERIAL

Airborne LiDAR has been useful in studying trees, whether it is for tree height estimation (Nilsson, 1996), retrieving biophysical variables (Popescu et al., 2004), or species classification (Yao et al., 2012). The LiDAR data collected for this paper was acquired 7 August 2009 by Riegl LMS-Q560, the study area is approximately 75 km east of Sault. Ste. Marie, Ontario, Canada. The average point density is approximately 40 points per m$^2$. 186 Individual trees crown was manually detected and segmented from the LiDAR scene before field validation. Field surveys were conducted in the summer of 2009 and 2011. There are eight field sites to capture the diversity of environmental conditions. Seven of the sites belong to a forested area and one site was chosen along two sides of a section of transmission corridor. From the 186 tree samples, 160 of them belong to the genera of *Pinus* (pine), 67 samples; *Populus* (poplar), 59 samples; and *Acer* (maple), 34 samples. The detailed description of the LiDAR dataset can be referred to our previous work presented in Ko et al. (2013). The 186 tree crowns are manually segmented out from the LiDAR scene as pre-processing. In this paper, we will focus on the classification of the three classes. Figure 1 show 5 example images for each genera, pine (top row), maple (middle row) and poplar (bottom row). Greyscale of the LiDAR points represents northing (m) in UTM.

## 3. METHODS

We separate our methods into four sections; section 3.1 described the preparation of input data and the method of multi-view data generation. Section 3.2 describes the splitting of training and testing dataset. Section 3.3 and section 3.4 describes the implementation of STN and MTN, respectively. STN classifies the trees into genera and MTN contain two tasks, first classifies the trees into genera and the second classifies the tree into coniferous or deciduous, using STN described in section 3.3 as base network.

### 3.1 Generating Multi-view LiDAR Tree Image Generation

For each segmented LiDAR tree, we produce 64 2-channel 2D images. The first channel (C1) of each image records the location of each detected LiDAR point and the second channel records the actual height of the tree. The first channel is generated by the inspiration of depth imageries that are widely used in computer graphics, where the depth channel records the distance between the object and the view point.

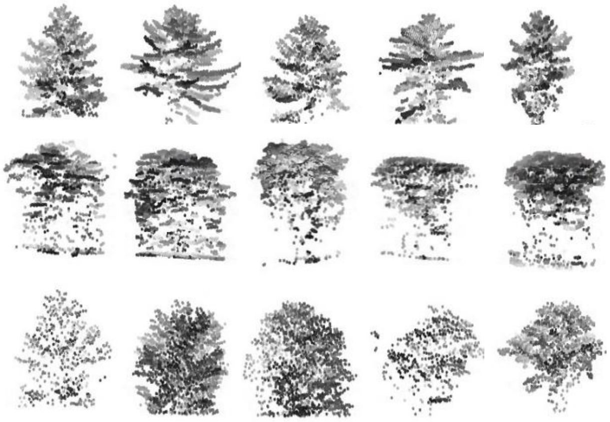

Figure 1. Example LiDAR segmented tree crown, the top row shows five examples of pine trees, the middle row shows five examples of maple trees and the bottom row shows five examples of poplar trees, the greyscale of the points represents UTM northing (m).

Similarly, we produced C1 by projecting each tree crown into UTM easting (x) and height (z). The representation of each discrete LiDAR reflection is symbolized by circle where the grey gradient (8-bit) represent the UTM northing (y). Each image has aspect x and z in the ratio of 1:1, the size of the image is fixed by adding a border around each image resulting a 500 × 500 pixels image. This implies the smaller trees (younger in age) and the bigger trees (older in age) will occupy the image the same way. Since C1 is normalized to the size of the image, we introduce the second channel (C2) where height above the lowest point of the tree crown (m) is recorded as pixel value. This way, two similar looking trees (in C1) that are differing in sizes will appear differently in C2, that is, the absolute size of the tree crown is preserved in C2. The images are then stored as 2-channel images. For illustration purpose, C1 is represented by red and C2 is represented by green. Figure 2(a) shows an example of a LiDAR pine tree projected to x-z axis. Each segmented tree crown has a local coordinate system where the origin (0,0) of the axis is located at the centre bottom of the tree crown. The grey scale represents the location of LiDAR points 3m out of the paper or 3m into the paper (-3m), this will lead to the generation of C1, represented in Figure 2(c). Figure 2(b) shows an example of the same pine tree with the same axes, where grey scale represents the height of the tree above the lowest point of the tree crown. Note that the highest point recorded for this tree is 12.3m. Figure 2(b) will produce C2; shown in Figure 2(d). By combining Figure 2(c) (red) and Figure 2(d) (green) will produce an example shown in Figure 2(e). All images are then resized to 48 × 48 pixels.

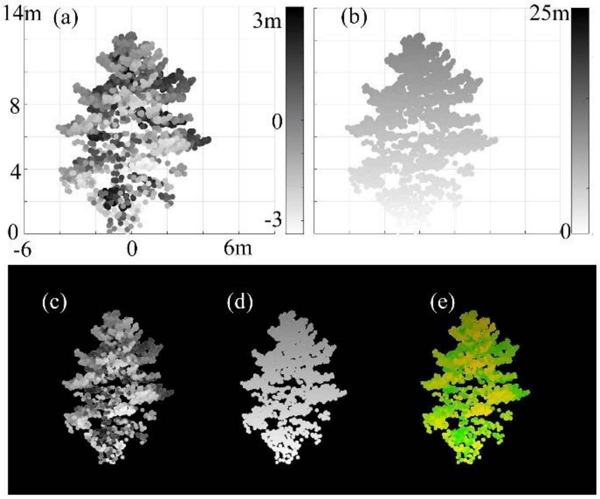

Figure 2. Example of a pine LiDAR tree projected on to x-z plane (a) shows the results when the origin represents the x-y centroid of the tree, the lowest point of the recorded LiDAR point is zero. Grey value represents the location of the LiDAR point out of the paper (+) and into the paper (-) (b) shows the results of the same tree on the same axes where grey scale represents height above the lowest recorded LiDAR point. (c) and (d) are the resulted image produced from (a) and (b), respectively. (e) is the result of combining (c) and (d) into a 2-channel image where (c) represents red channel and (d) represents green channel.

We then rotate the tree in 64 different angles about the vertical axis of the x, y centroid of the segmented tree crown. The angle of rotation is $2n\pi/64$ where $n = 1 \dots 64$, the projection plane remains the same where x represents the UTM-easting rotated and z represents the height rotated. Figure 3 shows an example of the generated 64 trees from a single pine tree. The top down view of the tree is included to indicate the 64 angles of rotation for the tree about the tree centre. This process is being repeated for all 160 trees.

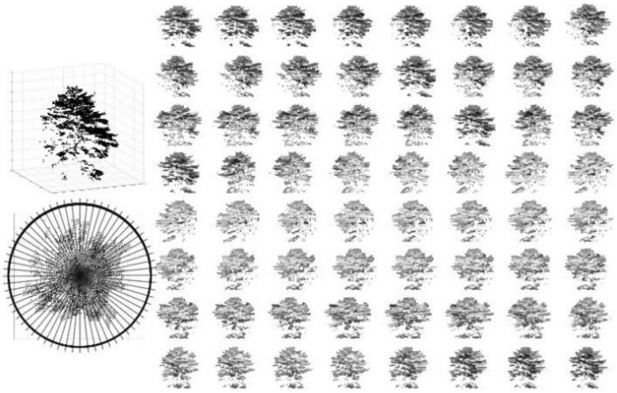

Figure 3. The generation of 64images from a single tree, example shows the results generated for C1

As a result of this method, we generate 10240 2-channel images (160 tree x 64 multiple views) for training and testing the results. Figure 3 illustrates the results of 64 C1 images generated from a single tree. 64 C2 images are generated the same way and are then combined into 2-channel images as the method described above.

### 3.2 Training Data and Testing Data

We split the data into two categories, training data (25%) and testing data (75%). We had chosen this partition from our previous study for comparison reasons (Ko et al., 2013). From the 160 trees we have collected, we randomly selected 40 trees for training, when a single tree is selected, 64 associated images are also selected as training. In the case of MTN, the same training and testing images are used for both tasks.

| | Training | | Testing | |
|---|---|---|---|---|
| *Genera (Class)* | *No. of trees* | *No. of images* | *No. of trees* | *No. of images* |
| M (D) | 8 | 512 | 26 | 1644 |
| Po (D) | 15 | 960 | 44 | 2816 |
| P (C) | 17 | 1088 | 50 | 3200 |

Where M = Maple; Po = Poplar; P = Pine; D = Deciduous; C = Coniferous

Table 1. Training and testing data partition for the experiments

### 3.3 Single-task Network (STN)

Motivated by LeNet-5 (LeCun et al., 1998), we designed a base network for tree genera classification, as in Figure 4. The base network consists of three Conv layers (Conv1, Conv2 and Conv3) and two fully connected layers (Fc1 and Fc2), where each Conv layer is composed of a series of layers of a convolution, batch normalization, Rectified Linear Unit (ReLU), max pooling, and dropout. Since this base network contains one single task, it is considered as a Single-task Network (STN). Note that for our experiment, the dimension of the input data is 48 pixels × 48 pixels × 2 channels.

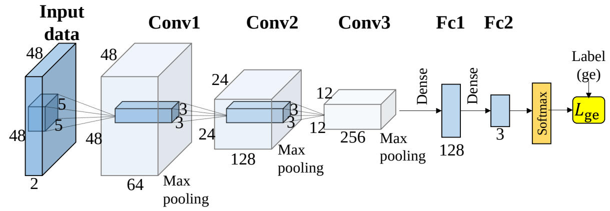

Figure 4. Summary of STN network

When the convolution operation is applied to the input with a filter, the result is called the convolution layer. The common two parameters for this layer are the filter sizes and the number of filters. In our study, the size of the filters is chosen based on examples such as LeNet-5 (LeCun et al., 1998) where authors have 32 × 32 pixel images, the first convolution operation use a 5×5 filter. In AlexNet (Krizhevsky et al., 2012), the input images are 224 × 244 pixel and the first convolution operation have filter size of 11 × 11. As our images are 48 × 48 pixels, we had chosen filter size of 5×5; 3×3 and 3×3 for all convolution layers 1, 2 and 3, respectively. Batch Normalization is the layer lies between convolution layers and Rectified Linear Unit (ReLU) layers and is used for speeding up the training (Ioffe et al., 2015). ReLU layer is the activation function where $f(x) = \begin{cases} x, x \geq 0 \\ 0, x < 0 \end{cases}$ which changes all the pixel values that are negative to zero. As a result, the stack of images only contains positive numbers. Dropout is a regularization technique for reducing overfitting in neural networks. It randomly drops units (along with their connections) from the neural network during training. Max Pooling layer reduce the size of the stacked images from the previous layer, where max pooling refers to recording the maximum value of the pixel within the window size. The pooled images will become the input of the next convolution layer. Dense, or full connections between nodes in two layers describes a fully-connected layer. The fully connected layer (Fc1 and Fc2) is a layer where the feature values vote for different classes in classification. The last step, the Softmax layer produces probabilities for the genera labels. We use the following equation to calculate loss function $L_{ge}(\boldsymbol{w})$:

Let $(\boldsymbol{x}_i, l_{ge,i})$, $i \in \{1, \cdots, N\}$ be the $i$th data in the training set, where $\mathbf{x}_i$ is the $i$th input data which is 48 pixels × 48 pixels × 2 channels and $l_{\mathrm{ge},i} \in \{1, \cdots, M\}$ is the label for $\mathbf{x}_i$ in the form of one-hot vector for the tree genera. The base network is trained to minimize the loss $L_{ge}(\boldsymbol{w})$ where:

$$L_{ge}(\boldsymbol{w}) = \frac{1}{N} \sum_{i=1}^{N} C(\boldsymbol{x}_i, l_{ge,i}; \boldsymbol{w}) \tag{1}$$

Where $C(\boldsymbol{x}_i, l_{ge,i}; \boldsymbol{w})$ = cross-entropy between predicted label probability and field-validated label
$\boldsymbol{w}$ = weights in the network

### 3.4 Multi-task Network (MTN)

Inspired by the work of Ruder (2017) and Liao et al. (2017), we extended the STN into MTN. In order to get the benefit from both tree genera classification and coniferous-deciduous classification, the proposed multi-task network is built to perform two tasks: 1) tree genera classification (major task), and 2) coniferous-deciduous classification (auxiliary task). Similar to STN described in section 3.3, MTN has the three convolution layers (Conv1, Conv2 and Conv3), with the difference of splitting into fully-connected layers that generate the outcome for each task. A summary of MTN is shown in Figure 5. The three convolution layers are shared among the two tasks Conv1, Conv2 and Conv3 with the same setting as STN described in Figure 4.

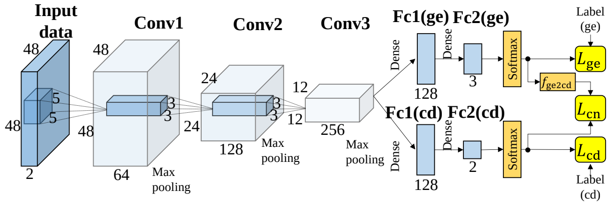

Figure 5. Summary of MTN network

On top of the loss $L_{ge}$ introduced in STN, we proposed two additional losses for training the MTN. The first one is $L_{cd}$, the loss for coniferous-deciduous binary classification (auxiliary task). It is defined as the average cross-entropy of the predicted probability for the classification and field validated labels. The second is $L_{cn}$, which we named as concurrence loss. The purpose of $L_{cn}$ is to minimize the inconsistencies between the outcome of the two tasks. We achieve the minimization by transforming the outcome for tree genera classification (major task) into coniferous-deciduous probabilities through the function $f_{\text{ge2cd}}$ (Figure 5). Then, we compared these derived probabilities with the probabilities obtained by the auxiliary task. The objective of the network is to minimize the difference by introducing $L_{\text{cn}}$, the average cross-entropy between these probabilities and the probabilities from the auxiliary task.

The proposed MTN is trained to minimize $L_{total}(\boldsymbol{w})$ which comprises of three losses $L_{ge}, L_{cd}(\boldsymbol{w})$ and $L_{cn}(\boldsymbol{w})$

$$L_{total}(\boldsymbol{w}) = L_{ge}(\boldsymbol{w}) + L_{cd}(\boldsymbol{w}) + L_{cn}(\boldsymbol{w}) \tag{2}$$

$$L_{ge}(\boldsymbol{w}) = \frac{1}{N}\sum_{i=1}^{N} \mathcal{C}(\boldsymbol{x}_i, l_{ge,i}; \boldsymbol{w}) \tag{3}$$

$$L_{cd}(\boldsymbol{w}) = \frac{1}{N}\sum_{i=1}^{N} \mathcal{C}(\boldsymbol{x}_i, l_{cd,i}; \boldsymbol{w}) \tag{4}$$

$$L_{cn}(\boldsymbol{w}) = \frac{1}{N}\sum_{i=1}^{N} \mathcal{C}(f_{ge2cd}(\boldsymbol{s}_{ge,i}), \boldsymbol{s}_{cd,i}; \boldsymbol{w}) \tag{5}$$

where $f_{ge2cd}(\boldsymbol{s}_{ge,i}) = \{s_{c,i}, s_{d,i}\}$
$s_{c,i} = \sum_{j \in C} \boldsymbol{s}_{ge,i}^{(j)}$
$s_{d,i} = \sum_{j \in D} \boldsymbol{s}_{ge,i}^{(j)}$
$\mathcal{C}(f_{ge2cd}(\boldsymbol{s}_{ge,i}), \boldsymbol{s}_{cd,i}; \boldsymbol{w})$ = cross-entropy between $f_{ge2cd}(\boldsymbol{s}_{ge,i})$ and $\boldsymbol{s}_{cd,i}$
$\boldsymbol{s}_{ge,i}$ = the M-dimensional vector, indicating the softmax output (predicted probability) for the tree genera classification of the $i$th data
$\boldsymbol{s}_{ge,i}^{(j)}$ = the $j$th element of $\boldsymbol{s}_{\text{ge},i}$, e.g. if $j = 1$ corresponds to 'maple', $\boldsymbol{s}_{\text{ge},i}^{(1)}$ is the probability of the $i$th input data being the 'maple'.
$\boldsymbol{s}_{cd,i}$ = 2-dimensional vector ($\boldsymbol{s}_{\text{cd},i}^{(1)}$ is for coniferous,$\boldsymbol{s}_{\text{cd},i}^{(2)}$ is for deciduous), indicating the softmax output (predicted probability) for the binary classification of the $i$th data

Both STN and MTN were implemented using Tensorflow Ver. 1.3 (Abadi et al., 2016) with NVIDIA GeForce GTX 1080 Ti. We are using Adam optimization (Kingma et al., 2014) with learning rate of 5.0 × $10^{-5}$, dropout at 0.7 and mini-batch size of 100 for both learning networks.

## 4. RESULTS AND DISCUSSION

We process the classification for both STN and MTN and provided the confusion matrices for the classification in section 4.2. The overall genera classification accuracy of STN is 88.7% and the overall genera classification accuracy of MTN is 91.0%. The improved classification accuracy indicates there is a future potential use of MTN for larger amount of data as well as increased number of genera. To illustrate the performance of MTN, Section 4.1 will show the results of the classification accuracies over various epoch for MTN and section 4.2 will shows the results of classification performance (STN and MTN) with confusion matrices.

### 4.1 Loss over Epoch for Multi-task Network

It has been shown in the previous research by Krizhevsky et al. (2012) that STN converges to minimum loss as epoch increases. One of our goals is to investigate if MTN will also perform the same way. To illustrate the performance of MTN, Figure 6 shows the loss functions

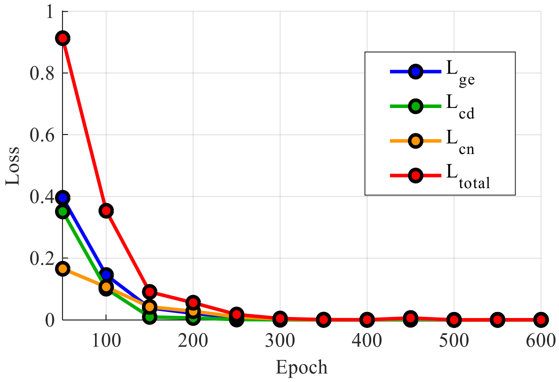

Figure 6. $L_{ge}$, $L_{cd}$, $L_{cn}$ and $L_{total}$ over Epoch

From the figure, we can see that all the loss functions ($L_{ge}$, $L_{cd}$, $L_{cn}$ and $L_{total}$) converge when epoch reached 300. Also, the loss for genera classification is higher than the coniferous-deciduous classification, meaning the binary classification generally has higher classification accuracy (a relatively easier task). Also, $L_{cn}$ decreases as epoch increases, meaning the inconsistencies between the predictions also decreases when epoch increases.

### 4.2 Confusion matrices for STN and MTN

To have a better understanding on the classification results, we present the confusion matrices of STN for genera classification and MTN for both coniferous-deciduous classification and genera classification. Table 2 shows the confusion matrix for genera classification for STN and Table 3 shows the confusion matrix for coniferous-deciduous classification for STN. Although the goal for STN is genera classification, we produce results for Table 3 to show classification accuracy improves for both our prime goal (genera classification) as well as the secondary goal (coniferous-deciduous classification). Table 4 shows the confusion matrix for genera classification for MTN and Table 5 shows the confusion matrix for coniferous-deciduous classification for MTN.

| | | Predicted | | | |
|---|---|---|---|---|---|
| | | *Maple* | *Poplar* | *Pine* | Omission error |
| **Measured** | *Maple* | 1553 | 76 | 35 | 0.07 |
| | *Poplar* | 0 | 2788 | 28 | 0.01 |
| | *Pine* | 171 | 555 | 2474 | 0.23 |
| Commission Error | | 0.10 | 0.18 | 0.02 | |
| | | Overall Accuracy = 88.7% | | | |

Table 2. Confusion matrix for genera classification for STN

| | | **Predicted** | | |
|---|---|---|---|---|
| **Measured** | | *Deciduous* | *Coniferous* | Omission error |
| | *Deciduous* | 4417 | 63 | 0.01 |
| | *Coniferous* | 726 | 2474 | 0.23 |
| Commission Error | | 0.14 | 0.02 | |
| Overall accuracy = 89.7% | | | | |

Table 3. Confusion matrix for coniferous-deciduous classification for STN

| | | **Predicted** | | | |
|---|---|---|---|---|---|
| **Measured** | | *Maple* | *Poplar* | *Pine* | Omission error |
| | *Maple* | 1538 | 67 | 59 | 0.08 |
| | *Poplar* | 0 | 2752 | 64 | 0.02 |
| | *Pine* | 115 | 384 | 2701 | 0.16 |
| Commission Error | | 0.07 | 0.14 | 0.04 | |
| Overall Accuracy = 91.0% | | | | | |

Table 4. Confusion matrix for genera classification for MTN with use of $L_{cn}$

| | | **Predicted** | | |
|---|---|---|---|---|
| **Measured** | | *Deciduous* | *Coniferous* | Omission error |
| | *Deciduous* | 4357 | 123 | 0.03 |
| | *Coniferous* | 499 | 2701 | 0.16 |
| Commission Error | | 0.10 | 0.04 | |
| Overall accuracy = 91.9% | | | | |

Table 5. Confusion matrix for coniferous-deciduous classification for MTN with use of $L_{cn}$

By comparing Table 2 and Table 4, we can see that the overall classification accuracy for genera increases from 88.7% to 91.0%. The omission error of pine decreases from 23% to 16% and the commission error of poplar decreases from 18% to 14% from STN to MTN. Although the classification of coniferous and deciduous is not our prime goal, the classification accuracy also increases from STN to MTN (89.7% to 91.9%). The omission error of coniferous trees decreases from 23% to 16% and the commission error of deciduous trees decrease from 14% to 10%.

We have some insights to the results obtained from Table 2 to Table 5. STN derived classification features that are designed explicitly for genera classification while MTN derived the share classification features for both tasks. The shared features are more generalized and perhaps had reduced some of the overfitting problems compared to STN and therefore has higher classification accuracy (Table 2 and Table 4). The improved classification accuracies for the coniferous-deciduous classification (Table 3 and Table 5) show that the proposed loss function $L_{cn}$ plays an important role in constraining the inconsistencies between the two tasks in MTN network. We think this constraint is particularly successful because of the nature of the hierarchical classification of tree genus.

## 5. CONCLUSION

Our paper has two major conclusions; the first is the use of existing CNN tools for processing LiDAR 3D data in 2D space. The advantage of such approach is that the classification features are automatically derived from the network and without human intervention. We had overcome the problem of insufficient training data by generating additional data though multi-view data augmentation. The second conclusion can be drawn from the successful results obtained from Multi-task Network (MTN). The genera classification accuracy had increased from 88.7% to 91.0%. The introduction of the constraint term $L_{\mathrm{cn}}$ has shown to be useful in improving classification accuracy. In the near future, we will obtain more LiDAR trees with higher number of classes and we would like to process a higher complexity set of data with the same MTN. This paper has provided us a strong foundation for the future work.

## ACKNOWLEDGEMENTS

We would like to acknowledge the NSERC (Natural Sciences and Engineering Research Council of Canada) and GeoDigital International Inc. (Alastair Jenkins) which have supported this research. Also, we would like to acknowledge the contributions of Professor Tarmo K. Remmel at Department of Geography, York University and Dr. Junjie Zhang (a former PhD student at York University), who supported the field surveying and data processing for this paper.

## REFERENCES

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D.G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y. and Zheng, X., 2016. TensorFlow: A system for large-scale machine learning. *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*. Savannah, Georgia, USA. pp.265-283.

Alonzo, M., Bookhagen, B., Roberts, D.A., 2014. Urban tree species mapping using hyperspectral and lidar data fusion. *Remote Sensing of Environment,* 148, pp. 70-83. doi:10.1016/j.rse.2014.03.018

Blomley, R., Hovi, A., Weinmann, M., Hinz, S., Korpela, I. and Jutzi, B., 2017. Tree species classification using within crown localization of waveform lidar attributes. *ISPRS Journal of Photogrammetry and Remote Sensing*, 133, pp. 142-156.

Boulch, A., Le Saux, B. and Audebert, N., 2017. Unstructured point cloud semantic labeling using deep segmentation networks. *Proceedings of the Eurographics Workshop on 3D Object Retrieval*, The Eurographics Association, Lyon, France, 23-34 April, pp. 17-24.

Brandtberg, T., 2007. Classifying individual tree species under leaf-off and leaf-on conditions using airborne lidar. *ISPRS Journal of Photogrammetry and Remote Sensing*, 61(5), pp. 325-340.

Chatfield, K., Simonyan, K., Vedaldi, A., Zisserman, A., 2014. Return of the Devil in the Details: Delving Deep into Convolutional Nets. *Proceedings of the British Machine Vision Conference 2014*. doi:10.5244/c.28.6

Girshick, R., 2015. Fast R-CNN. *In Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440-1448.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep learning. MIT Press, Cambridge, MA.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. doi:10.1109/cvpr.2016.90

Holmgren, J., and Persson, Å., 2004. Identifying species of individual trees using airborne laser scanner. *Remote Sensing of Environment*, 90(4), pp. 415-423.

Holmgren, J., Persson, Å., and Söderman, U., 2008. Species identification of individual trees by combining high resolution LiDAR data with multi-spectral images. *International Journal of Remote Sensing*, 29(5), pp.1537-1552.

Huang, J. and You, S., 2016. Point cloud labeling using 3d convolutional neural network. *Proceedings of the International Conference on Pattern Recognition, IEEE*, Cancun, Mexico, 4-8 December, pp. 2670-2675.

Jones, T.G., Coops, N.C., and Sharma, T., 2010. Assessing the utility of airborne hyperspectral and LiDAR data for species distribution mapping in the coastal Pacific Northwest, Canada. *Remote Sensing of Environment*, 114(12), pp. 2841-2852.

Kingma, D. and Ba. J., 2015. Adam: A method for stochastic optimization. *In the International Conference on Learning Representations (ICLR 2015)*, San Diego, 2015. arXiv:1412.6980.

Ko, C., Sohn, G. and Remmel, T.K., 2013. A spatial Analysis of Geometric Features Derived from High-Density Airborne LiDAR Data for Tree Species Classification. *Canadian Journal of Remote Sensing*. 39(s1), pp.S73-S85.

Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. ImageNet Classification with Deep Convolutional Neural Networks, In Advances in Neural Information Processing Systems 25, pp. 1106-1114.

Lawin, F.J., Danelljan, M., Tosteberg, P., Bhat, G., Khan, F.S. and Felsberg, M., 2017. Deep projective 3d semantic segmentation, pp. 1-12.

Lawin, F.J., Danelljan, M., Tosteberg, P., Bhat, G., Khan, F.S. and Felsberg, M., 2017. Deep projective 3d semantic segmentation, pp. 1-12.

Liao, H. and Luo, J., 2017. A Deep Multi-task Network Approach to Skin Lesion Classification. *Joint Workshop on Health Intelligence* February 4, 2017 - February 10, 2017.

Lim, K., Treitz, P., Wulder, M., St-Onge, B., and Flood, M., 2003. LiDAR remote sensing of forest structure. *Progress in Physical Geography*, 27(1), pp.88-106.

LeCun, Y., Bottou, L. Bengio, Y. and Haffner, P., 1998. Gradient-based learning applied to document recognition, *Proc. IEEE*, 86(11), pp.2278-2324.

LeCun, Y., Bengio, Y. and Hinton, G., 2015. Deep learning. *Nature*, 521(7553), pp. 436-444.

Liu, L., Pang, Y., Fan,W., Li, Z., and Li, M., 2011. Fusion of airborne hyperspectral and LiDAR data for tree species classification in the temperate forest of northeast China. *Geoinformatics, 2011 19th International Conference on Geoinformatics,* pp. 1-5.

Nilsson, M., 1996. Estimation of tree heights and stand volume using an airborne lidar system. *Remote Sens. Environ*. 56, pp.1-7.

Popescu, S.C., Wynne, R.H. and Scrivani, J.A., 2004. Fusion of small-Footprint lidar and multispectral data to estimate plot-Level volume and biomass in deciduous and pine forests in Virginia, USA. *For. Sci*, 50, pp. 551-565.

Ramsundar, B., Kearnes, S., Riley, P., Webster, D., Konerding, D., Pande, V., 2015. Massively multitask networks for drug discovery. arXiv preprint arXiv:1502.02072.

Ruder, S., 2017. An overview of Multi-task Network in deep neural networks. arXiv preprint. arXiv:1706.05098.

Su, H., Maji, S. Kalogerakis, E. and Learned-Miller E., 2015. Multi-view convolutional neural networks for 3d shape recognition. *In Proceedings of the IEEE International Conference on Computer Vision*, pp. 945-953.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* doi:10.1109/cvpr.2015.7298594

Yao, W., Krzystek, P., and Heurich, M., 2012. Tree species classification and estimation of stem volume and DBH based on single tree extraction by exploiting airborne full-waveform LiDAR data. *Remote Sensing of Environment*, 123, pp. 368.