Southern California Edison LiDAR

Suarsana, Carissa and Lu, Wilson

2023

**Problem Statement**

Senate Bill 901 (SB-901), passed in 2018, requires utility companies to have wildfire-mitigation plans. Utility companies are required to visually inspect their assets every 12–24 months (about 2 years) of service and thoroughly inspect them every 3–5 years based on their type. Generally, these assets require many hours of management and are not easily accessible due to their locations. Dispatching teams of people to assess the status of power equipment is expensive and time consuming.

Southern California Edison (SCE) currently relies on third party vendors to process Light detection and ranging (LiDAR) point-cloud data captured for many purposes including the identification of assets in the service territory and improving these assets’ geolocation data. The development of a LiDAR data processing tool at SCE offers the potential to eliminate the need for costly vendors by bringing the processing in-house, allowing SCE to have overall better control.

**Objectives**

The main objective of this project is to develop a proof of concept for a LiDAR point-cloud data-processing tool to test SCE’s ability to bring the process in-house and study the potential benefits associated with such a tool. LiDAR is used for mass asset surveys to identify components and configurations on each pole, vegetation management to survey trees infringing on distribution lines, and structural load analysis to determine the number of poles in high-fire-risk areas. The SCE Asset Strategy & Planning team is working on improving asset-data quality, identifying solutions to mitigate data-quality issues, and forecasting fire hazards. LiDAR offers the opportunity to extract more accurate geolocation data for assets in the service territory. This project offers a first use case in studying its potential benefits.

**Background and Business-Case Justification**

LiDAR, a remote-sensing technology, uses pulsed lasers to measure variable distances, heights, or depths of objects and areas. It accurately, precisely, and flexibly examines natural and artificial environments. LiDAR data are generally collected aerially or terrestrially using an unmanned aerial vehicle (UAVs) or unmanned ground vehicles (UGVs). UAVs are remotely operated to scan areas of interest from altitudes greater than ten meters. At a minimum, this process requires a two-person team to remotely operate the UAV and verify the data are correct (National Ocean Service, 2021). Software can read these point-cloud data for further processing. In contrast, UGVs detection distances range below ten meters to perform precise geometric measurements. UAVs and LiDAR data provide several benefits over sending people to physically inspect all assets of interest. For instance, a UAV can easily scan large areas without regard to terrain (steep slopes, dense forests, etc.).

Several studies have examined the extraction of objects from point-cloud data. For instance, Van Leeuwen and Nieuwenhuis (2010) examined the current and future potential for leveraging LiDAR data to assess and manage forest structures, specifically how remote sensing and classification can identify specific trees in clusters and more closely identify species. The article is relevant to this use case because this study examines whether LiDAR can be used to identify power poles and structures, which may be imbedded in forests or other rural areas. Van Leeuwen and Nieuwenhuis demonstrate that remote sensing techniques may help identify objects in a forest (in their case, individual trees) and conclude that further research is needed to assess remote sensing and forest management, as well as using models to recognize objects within point-cloud data. Power poles and towers may blend into a forest canopy similar to individual trees.

In 2009, Prokhorov (2009) examined how 3D LiDAR imaging could be used in conjunction with a recurring neural network (RNN) to identify different objects. With the progression of scanners, 3D LiDAR images provide enhanced measurement data (Prokhorov, 2009). Prokhorov investigated how the space of points between various objects could be leveraged to create a model to recognize objects. This research concluded that the RNN model showed promise, and that further research into training RNN models is warranted, as is pursuing better 3D data.

Maggiori et al. (2019) created an end-to-end framework to classify satellite imagery using convolutional neural networks (CNNs). In their study, they observed how CNN has significant advantages when classifying satellite imagery data to identify objects and produce quality imagery. However, they also noticed that untrained models did not perform as well. They leveraged an existing model and constructed a set of manually classified data and saw significant improvement in the model. Therefore, they propose a two-step approach leveraging a small set of manually classified data to train a model to classify a large set of unclassified data.

Kudinov (2019), in collaboration with ESRI and AAM Group, used the point convolution neural network (PointCNN) framework to automatically identify power lines and poles. The group used artificial intelligence for the labor-intensive task of manually labeling the point cloud. Their study area was a city in Australia, and their dataset contained around 540 million points. They trained their PointCNN model using four classes: other, wires, stray wires, and utility poles to successfully identify power poles.

Fan et al. (2021) studied the You Only Look Once (YOLO) deep-learning algorithm to detect objects in point-cloud datasets. The focus of their research was object detection for self-driving vehicles. These vehicles need real-time information to make decisions and avoid collisions. Consequently, the researchers propose an alternative computationally efficient algorithm dubbed LS-R-YOLOv4 using color images and point-cloud data to precisely segment and detect objects. Borcs et al. (2017) proposed a pipeline that quickly classifies point clouds. One component of this pipeline is a CNN trained to classify objects. The model supports the identification of vehicles and pedestrians in urban settings.

Brubaker et al. (2013) showed that LiDAR data can be used to accurately pinpoint micromorphology of a large area and compared their results to field-surveyed plots to determine their accuracy. They compared a digital elevation model (DEM) generated from LiDAR data to the surveyed plots. From their findings, they were able to learn that their research was accurate to within 0.3–0.4 m based on the manual survey, which is accurate up to a single point in the point cloud. Their data allowed them to generate the surface constraint of the surveyed area faster and from a greater distance compared to a traditional survey. The DEM is important as it allows LiDAR data to be accurately separated from ground, water, or any surface constraints based on elevation.

Azevedo et al. (2019) showcased the use of UAVs to replace helicopters due to their risks and associated costs. UAVs and LiDAR have lower equipment costs over time, as a team of just a few people can ensure that the data is correct and control the UAV. Equipped with the proper sensors, the UAV can quickly scan a large area and send data back to the controller. From there, the LiDAR data can be converted to point-cloud data and fed through an algorithm and software to help identify and sort items in the LiDAR data. They argue that, while the algorithm they used failed to correctly identify possible points, those points were classified as unidentified due to the difficulty of differentiating between vegetation and other sources. They conclude that a more powerful algorithm may correctly identify the points of interest and that graphics processing units (GPUs) can be used to reduce the time required to process the raw data.

Nahhas et al. (2018) proposed machine learning with LiDAR data and orthophotos. They showed that the CNN algorithm was able to transform, organize, and label the data. With the orthophotos and LiDAR data, they created a digital surface model, DEM, and shapes. They also input other data to detect buildings. From their findings and experiments, the CNN and machine-learning model accurately classified background and buildings up to a single data point and drew the geometry and shapes of the building from the LiDAR and orthophotos. Using this model, they were able to transform low-level detail into highly detailed, classified features.

Sultan et al. (2022) empirically focused on machine learning to identify power poles and towers from point-cloud data. This study sought to demonstrate the use of a deep-learning model developed by Azevedo et al. (2019) to determine whether deep learning is a viable solution for identifying power assets in three California areas. This study instantiated an existing trained model to determine whether deep learning is an effective solution for extracting the desired objects from point-cloud data. The deep-learning model successfully identified power poles in both rural and urban areas. However, the model performance was better in urban areas than in rural areas. This study supports the literature that deep learning can successfully classify point clouds. To improve the model performance and to ensure optimal results when training the model, authors suggest more accurately labeled data representing the objects of interest.

LiDAR data serves as a cost-efficient alternative for surveying large areas of land and generating real-time images of objects on the ground. The point-cloud data generated by scans can be analyzed to identify assets in need of maintenance. In addition to the efficiency afforded by LiDAR, utility companies can potentially lower labor and transportation costs as there is no need to unnecessarily send maintenance crews into the field. The cost of LiDAR depends on the type of equipment to be purchased and the range and scope of work (Antunes, 2018). LiDAR drones can potentially be cost effective in difficult-to-reach forested areas, rural towns, or high elevations. At the same time, LiDAR can be used in high-density areas such as urban or suburban areas (Singh et al., 2015). The high upfront cost leaves just maintenance of the equipment, future upgrades, and pilot licensing as needed (Van Tassel, 2021). These costs can be calculated in advance, while the ongoing costs of dispatching workers depend on the scope of work and may not be easily estimated due to fluctuating rates of pay (Glavinich and Chichester, 2021). In many cases, contractors may need to be hired in areas that are difficult to reach and may not have the exact quality control utility companies need. However, manually assessing and inspecting equipment is beneficial as the information about them can be updated in real time, whereas LiDAR data must be processed and analyzed to ensure the data is error free (Azevedo et al., 2019). A high-scale scan must be performed of target areas to produce error-free point-cloud data and these data must be processed to ensure assets are correctly identified (Nahhas et al., 2018). LiDAR technology provides several benefits when surveying objects. Therefore, this study sought to answer the following question.

**Can Southern California Edison process LiDAR point-cloud data to accurately define asset locations?**

The literature suggests deep learning can be used to classify objects of interest. As a result, this study will instantiate the deep-learning model deployed by Sultan et al. (2022) to determine its effectiveness at processing SCE’s point cloud data. In addition, other ArcGIS Pro classification tools will be studied and tested to gauge their effectiveness at classifying poles and towers. This study may be of interest to SCE’s executive team to bring the entire process in-house, study the potential benefits and improve the accuracy of assets’ geolocation data.

**Data**

**Table 1:** Point-cloud datasets

|  |  |  |
| --- | --- | --- |
| **Point-cloud datasets** | **File size** | **LAS Points** |
| SCE LiDAR data | 7.83 GB | 233,571,578 |
| test (Training Data) | 38 MB | 3,253,697 |
| val\_small (Training Data) | 38 MB | 1,233,135 |
| train\_small (Training Data) | 106 MB | 4,042,521 |

**Methodology**

This research study will explore ArcGIS geoprocessing tools, including the deep-learning model, image analytics, and other tools that complement ArcGIS. The goal of this exploration is to determine the costs and benefits of SCE performing its own point-cloud analysis instead of paying for outside vendors to perform this service.

ArcGIS Pro provides three tools to classify data, train a model, and use a model for Point Cloud data classification. SCE has access to all these tools under their current License from the Environmental Systems Research Institute (ESRI). The following ArcGIS Pro classification tools will be explored and tested by the project team:

1. Classify LAS Ground

2. Classify LAS Building

3. Classify LAS by Height

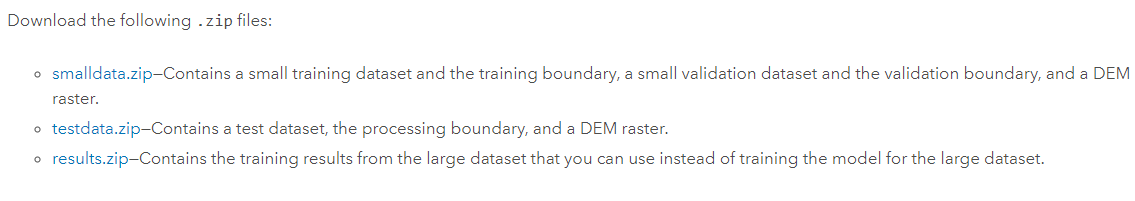
4. Classify LAS Noise

5. Change LAS Classification Codes

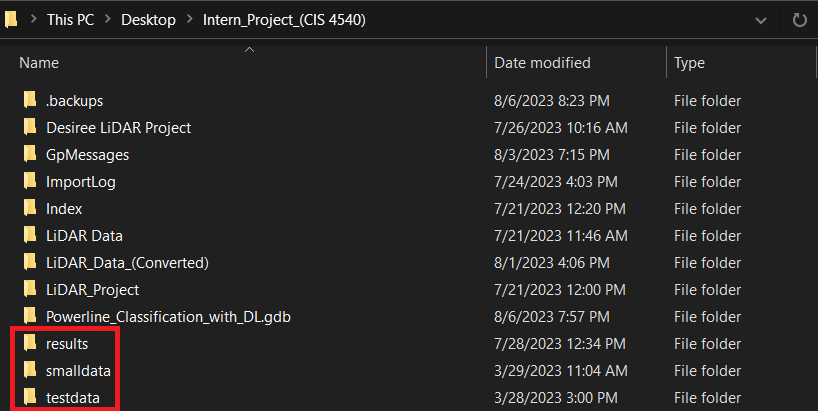
Phase 1 of this project will include exploration of the tools to evaluate whether ArcGIS LAS classification tools will support the classification of power poles and towers, the development of the cost–benefit analysis and making a preliminary recommendation. Phase 2 of the project may include training the model on SCE’s private point-cloud data to give it the best chance of correctly identifying buildings and electrical-system assets in SCE’s service territory.

**1.0 Download Training Data**

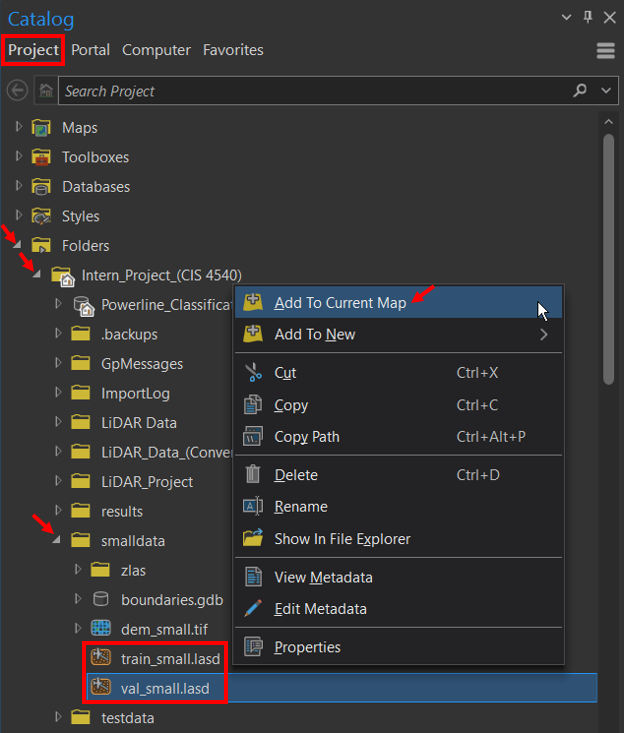
* **Step 1**: Navigate to the Classify power lines using deep learning tutorial, available at: <https://learn.arcgis.com/en/projects/classify-powerlines-from-lidar-point-clouds/>
* **Step 2**: Scroll down to the "**Download the Data**" section and download the .zip files (smalldata, testdata, results).



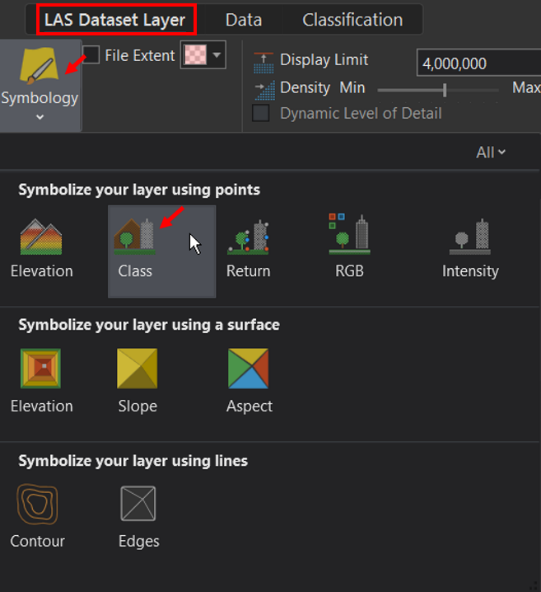
* **Step 3**: Extract the files into a folder that you can easily find/access (preferably the same folder you created your project in).



* **Step 4**: Open ArcGIS Pro, in the **Catalog** pane, click the **Projects** tab, expand the **Folders** folder, expand your project folder, expand the **smalldata** folder, and add **train\_small.lasd** and **val\_small.lasd** by right clicking on the files and selecting “**Add to current map**.”

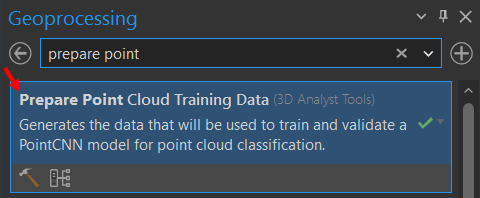


* **Step 5**: In the **Contents** pane, click the **train\_small.lasd** layer to select it. On the ribbon, click the **LAS Dataset Layer** tab. In the **LAS Dataset Layer** tab, under the **Drawing** section, for **Symbology**, click the drop-down menu and choose **Class.** Do the same for **val\_small.lasd**.

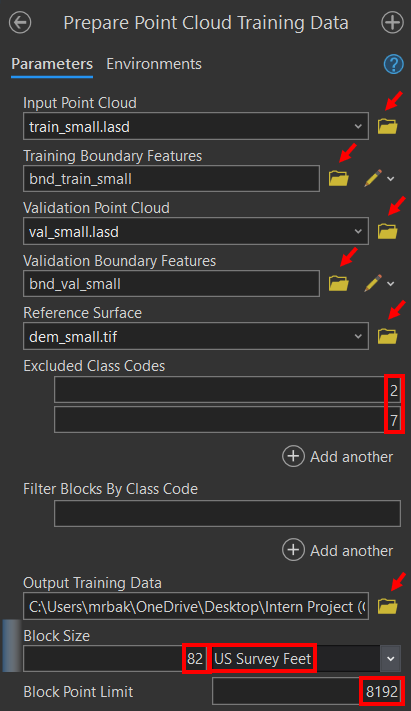


**1.1 Preparing Point Cloud Training Data**

* **Step 1**: Navigate to the Geoprocessing Tool pane, search and open **Prepare Point Cloud Training Data**.

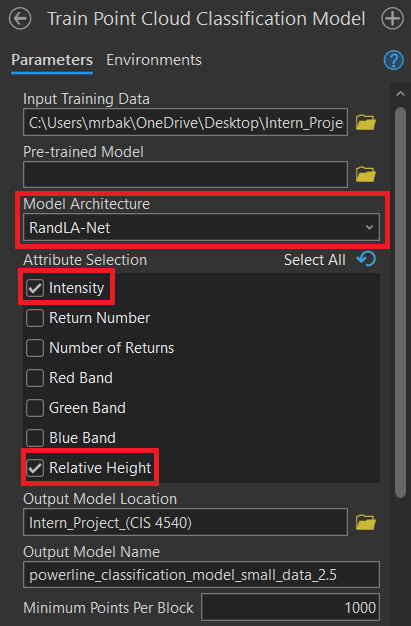


* **Step 2**: For **Input Point Cloud**, choose **train\_small.lasd**. For **Training Boundary Features**, click **Browse**, browse to the **smalldata** folder, open **boundaries.gdb**, and choose **bnd\_train\_small.** For **Validation Point Cloud**, choose **val\_small.lasd**. For **Validation Boundary Features**, click **Browse**, browse to **smalldata** folder, open **boundaries.gdb**, choose **bnd\_val\_small**. For **Reference Surface**, click **Browse,** browse to **smalldata** folder, click **dem\_small.tif**.
* **Step 3**: Exclude Class Codes **2** (ground) and **7** (noise) in **Excluded Class Codes** section. This will allow the model to run faster.
* **Step 4**: For **Output Training Data**, click **Browse** and browse to your **results** folder. Enter **training\_data\_small.pctd** for **Output Training Data**.
* **Step 5**: Enter **82** for **Block Size**. For **Unknown**, click the drop-down menu and choose **US Survey Feet**. For **Block Point Limit**, keep the default value of **8192**.

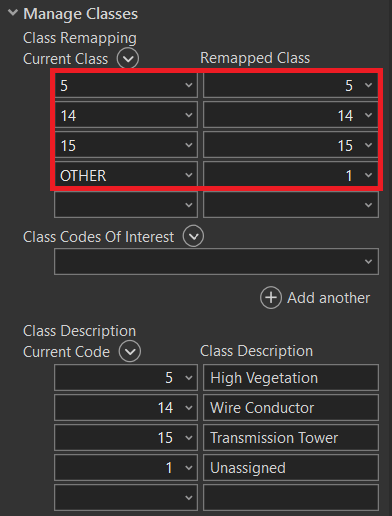


**1.2 Training Classification Model**

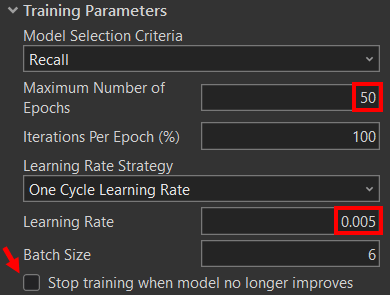
* **Step 1**: Navigate to the Geoprocessing Tool pane, search and open **Train Point Cloud Classification Model**.
* **Step 2**: For **Input Training Data**, click **Browse**, browse to **results** folder and select **training\_data\_small.pctd**. Keep the default option of **RandLA-Net** for **Model Architecture.**
* **Step 3**: Check **Intensity** and **Relative Height** for **Attribute Selection.**
* **Step 4**: For **Output Model Location**, select your **results** folder and give the model a name in the **Output Model Name** section. Leave the **Min. Points Per Block** at the default value of **1000**.



* **Step 5**: Add classes **5**, **14**, and **15** for **Current Class**. Reassign all **OTHER** class codes to **1**.

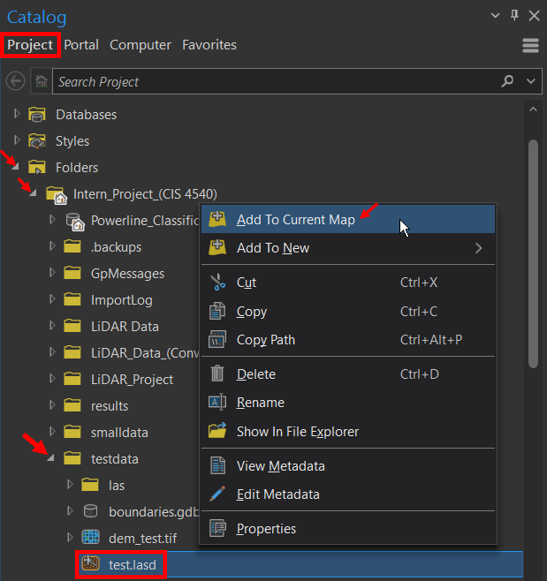


* **Step 6**: Change **Max. Number of Epochs** to **50**. Leave **Iterations Per Epoch (%)** at the default value of **100**.
* **Step 7**: Change **Learning Rate** to **0.006**. Leave **Batch Size** at the default value of **6** and uncheck "**Stop training when model no longer improves**".

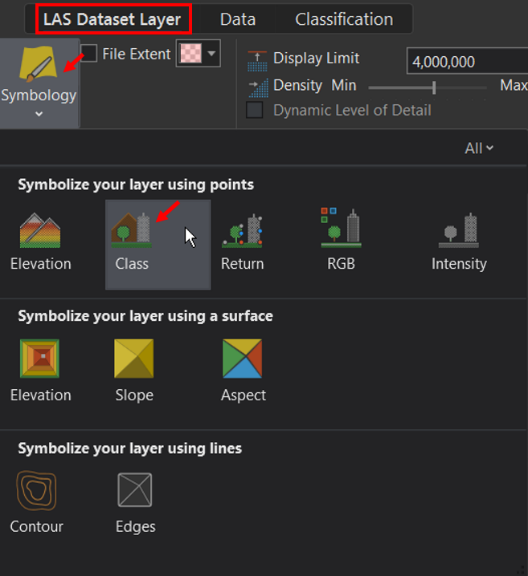


**1.3 Testing Classification Model on Training Data**

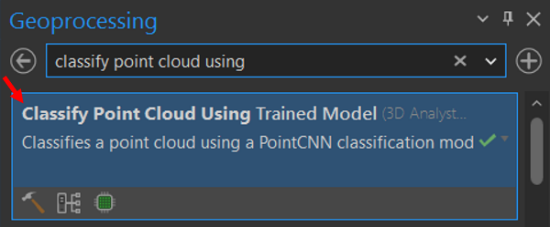
* **Step 1:** Open ArcGIS Pro, in the **Catalog** pane, click the **Projects** tab, expand the **Folders** folder, expand your project folder, expand the **testdata** folder, and add **test.lasd** by right clicking on the files and selecting “**Add to current map**.”



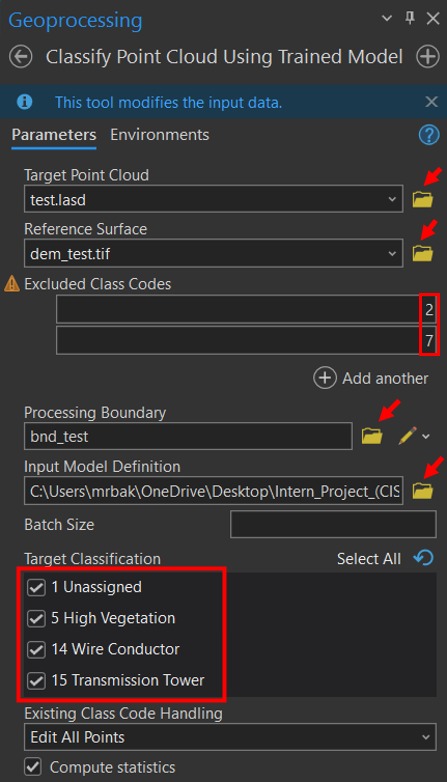
* **Step 2**: In the **Contents** pane, click the **test.lasd** layer to select it. On the ribbon, click the **LAS Dataset Layer** tab. In the **LAS Dataset Layer** tab, under the **Drawing** section, for **Symbology**, click the drop-down menu and choose **Class.**



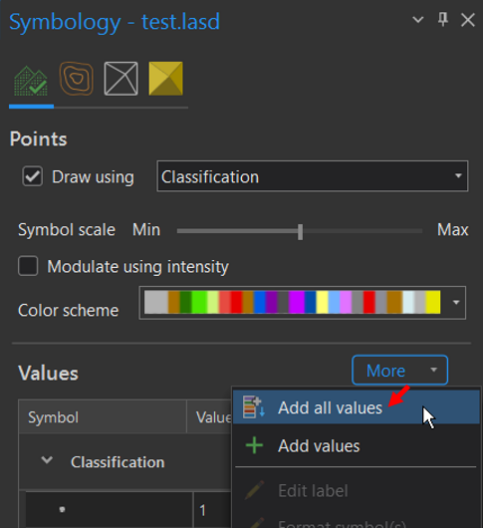
* **Step 3**: Navigate to the Geoprocessing Tool pane, search, and open **Classify Point Cloud Using Trained Model**.



* **Step 4**: In the **Classify Point Cloud Using Trained Model** window, select **test.lasd** as the **Target Point Cloud**. For **Reference Surface**, select **dem.tif**. For **Processing Boundary,** browse to **testdata\boundaries.gdb**, and select **bnd\_test**. Exclude classes **2** and **7** in the **Excluded Class Codes** section.
* **Step 5**: For **Input Model Definition**, click **Browse** and navigate to the **EMD file** in the data folder created in **Section 1.2** for the epoch with the highest recall percentage for Class code 14, Transmission Wire. Ensure all 4 classes are selected for **Target Classification**.

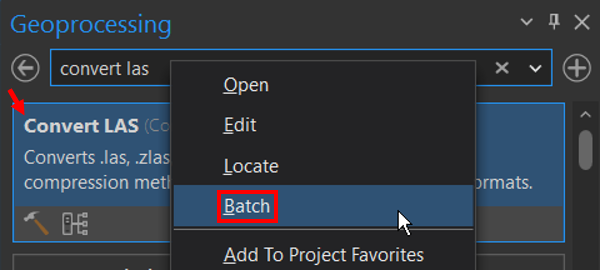


* **Step 6**: Once the tool is finished running, click **test.lasd** in the **Contents** pane. In the **Symbology** pane, next to **Value**, click **More** and click “**Add all values**.”

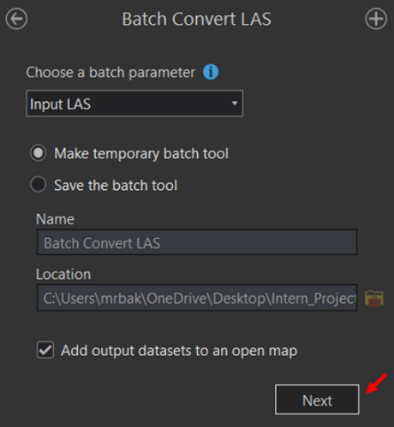


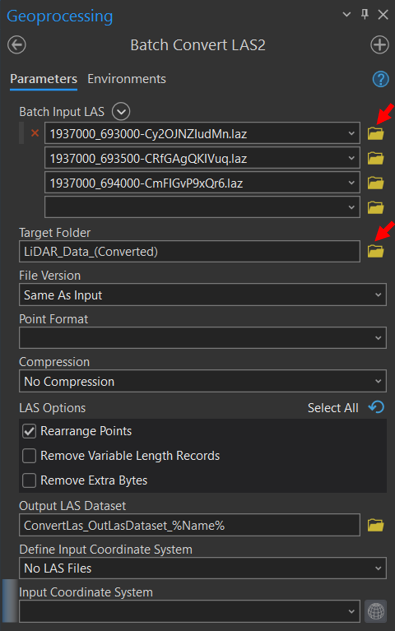
**1.4 Convert LAZ to LAS**

* **Step 1**: Navigate to the Geoprocessing Tool pane and search for **Convert LAS**. Right click on it and click **Batch**.



* **Step 2**: The following window will appear. Do not change any of the options and simply click **Next** to proceed to the next step.



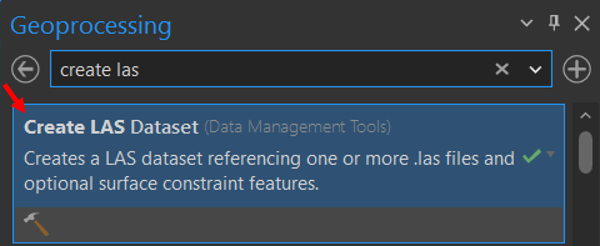
* **Step 3**: For the **Batch Input LAS**, click the **Browse** icon and navigate to the folder where you saved your .laz files. Proceed to add all the files you need to convert. 
* **Step 4**: Select a **Target Folder** to store your converted data. We recommend creating a separate folder from your original data folder to store the converted data. Leave all other parameters as the default ones selected.



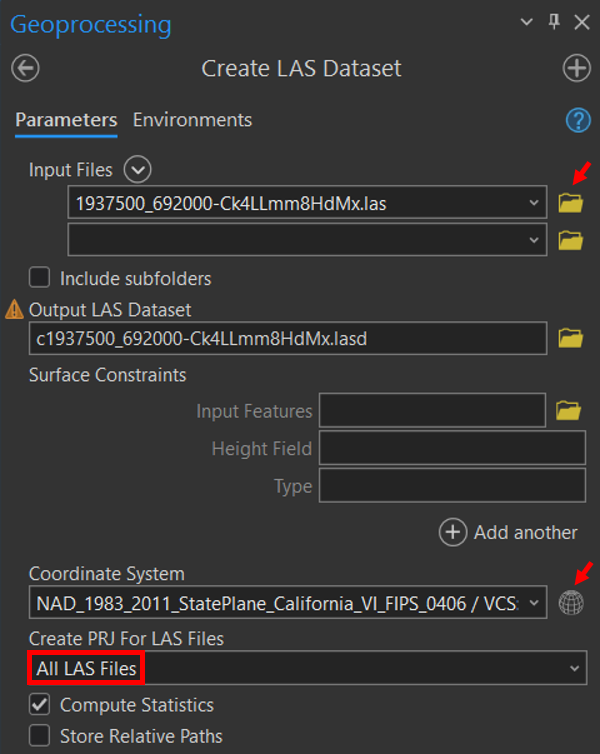
* **Step 5**: Add all .las files to your project by right clicking on a .las file in the **Catalog** pane and selecting the option, “**Add to Current Map**.”
* **Step 6**: In the **Contents** pane, click a .las layer to select it. On the ribbon, click the **LAS Dataset Layer** tab. In the **LAS Dataset Layer** tab, under the **Drawing** section, for **Symbology**, click the drop-down menu and choose **Class**.

**1.5 Create LAS Dataset**

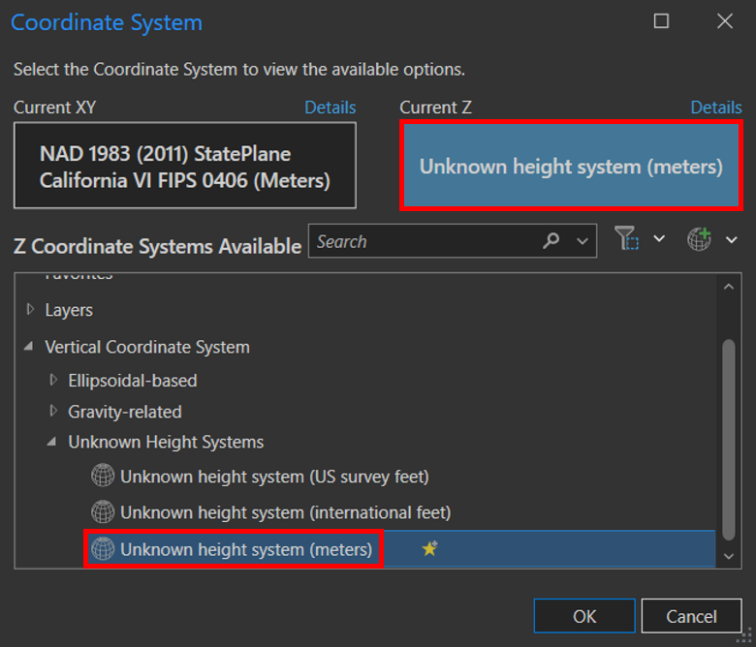
* **Step 1**: Navigate to the Geoprocessing Tool pane, search and open **Create LAS Dataset**.



* **Step 2**: In the **Create LAS Dataset** window, select a .las file as the **Input Files**. The **Output LAS Dataset** field will automatically generate a name for the LASD.



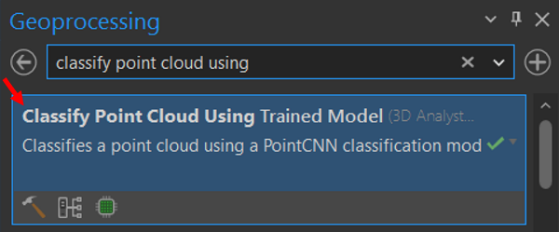
* **Step 3**: In the **Coordinate System** section, click on the **Globe** icon and confirm that there is a **Current Z** system selected in the same unit as the **Current XY** (Meters in our case). When in doubt regarding the **Current Z** system, select an **Unknown height system** in the appropriate units.



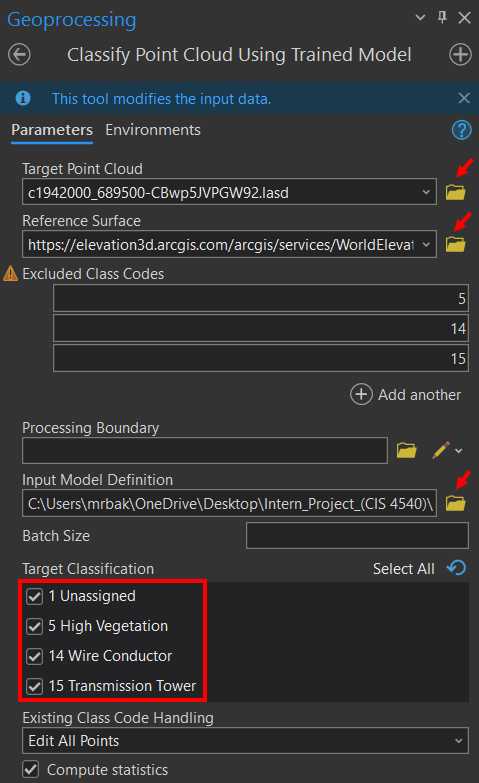
* **Step 4**: Select “**All LAS Files”** in the drop-down menu for **Create PRJ For LAS Files** field.
* **Step 5**: Once the tool is finished running, click a .**lasd** in the **Contents** pane. In the **Symbology** pane, next to **Value**, click **More** and click “**Add all values**.”

**1.6 Classify LASD with Classification Model**

* **Step 1**: Navigate to the Geoprocessing Tool pane, search, and open **Classify Point Cloud Using Trained Model**.

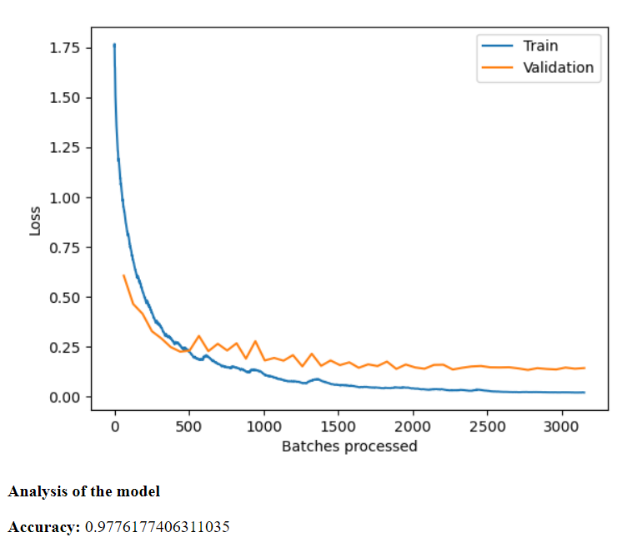
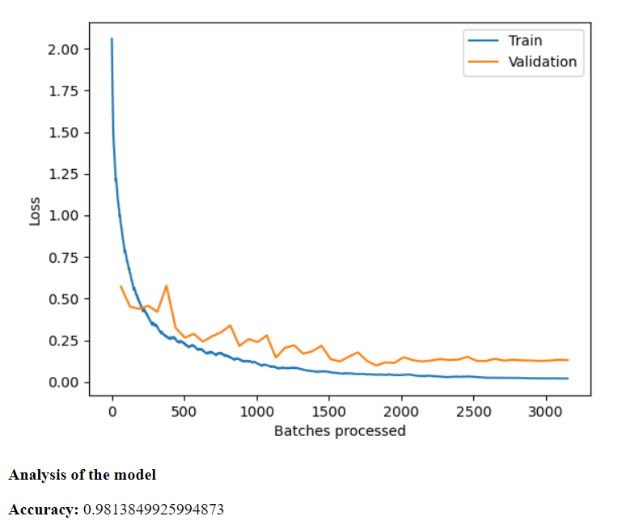


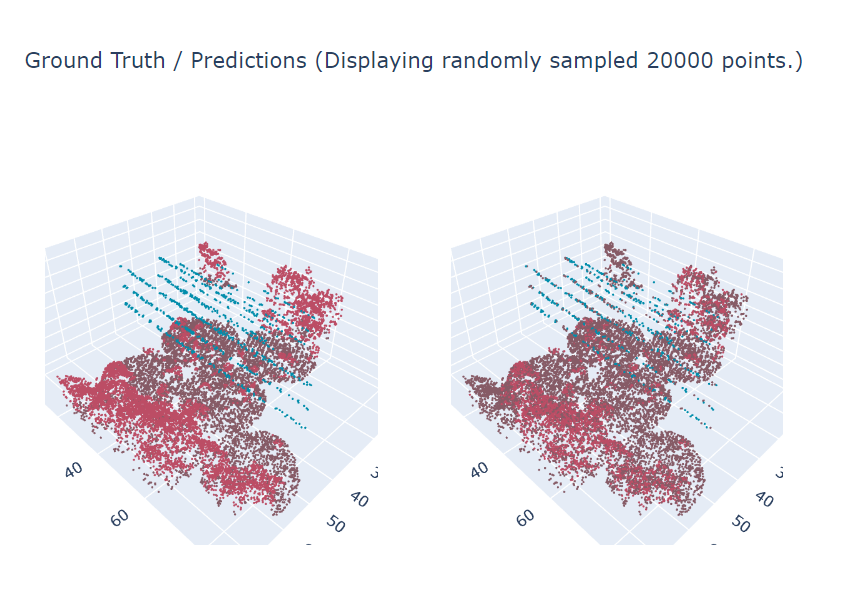
* **Step 2**: In the **Classify Point Cloud Using Trained Model** window, select a .lasd file as the **Target Point Cloud**. For **Reference Surface**, in the drop-down menu, select “**WorldElevation3D/Terrain3D**.”
* **Step 3**: Exclude classes **5**, **14**, and **15** in the **Excluded Class Codes**.
* **Step 4**: For **Input Model Definition**, click **Browse** and navigate to the **EMD file** in the data folder created in **Section 1.2** for the epoch with the highest recall percentage for Class code 14, Transmission Wire. Ensure all 4 classes are selected for **Target Classification**.



**Results**

After successfully running the “Train Point Cloud Classification Model” tool in ArcGIS, the tool generated results on the accuracy and precision of our classification model in the form of graphs and excel sheets. The more we trained our classification model, the less training and validation loss occurred which resulted in faster times to train the time, but also a better recall rate for each classification class. As shown in the graphs below, the result on the left was from one of our initial runs for training our model. The accuracy of our classification model was relatively high at 97.76%. On the right are the results of our training model after rerunning it with the same set of training data and parameters. While the results of our training model increased in the second run, the increase wasn’t drastic, going from 97.76% to 98.14%.

 0 In this picture generated after running our most recent classification model, it showed the ground truth, which represents the actual point data from our dataset, versus the model’s predictions. The ground truth is the left most square of data while the predictions are the right most square of data in the image. We observed that our model’s predictions were very similar to that of the ground truth but noticed that some of the point data in our model classified some points incorrectly when compared to the ground truth. In the bottom left corner of the predictions square, we observed that some of the data was classified as a different color compared to the same points on the ground truth. We believe that this difference in color in our predictions model could potentially explain why our classification model wasn’t able to classify the transmission towers and poles in the correct classification code.



The CPCUTM (Classify Point Cloud Using Trained Model) geoprocessing tool did not successfully classy point-cloud data points as transmission poles and towers. The model was able to identify and classify high vegetation and transmission wires in most, if not all, the LAS datasets. However, the deep-learning training model developed through the ArcGIS Online tutorial was unable to correctly classify all classification codes. Results of these areas are shown in the figures below.

A computer generated image of a forest

Description automatically generated

While the tools technically did not accomplish the objective of identifying transmission poles and towers, the model can still be retrained to help achieve this. The tool performed better as it was retrained, but it still had difficulty finding most of the transmission poles and towers within the datasets analyzed.

Moreover, the deep-learning model did not correctly classify any transmission poles and towers within the classified datasets.

The remote-sensing classification tools did result in increased performance of the model compared to the unclassified data. With both increased classification intensity and unclassified data, the model did produce differing results. However, it still failed to correctly classify a large majority of transmission poles and towers in the service territory. The model’s general results were underwhelming. The results below demonstrate that the model should have classified the transmission tower formed and look like the image shown previously of the training model.

LASD Before Classification Model



LASD After Classification Model

A wind turbines in the air

Description automatically generated

**Discussion of findings**

Based on our analysis and efforts to convert LAZ files to LAS files and use the CPCUTM geoprocessing tool in combination with our deep-learning training model, our project generated several findings.

The LAZ-to-LAS file conversion led us to discover a batch conversion tool built in the laszip tool. The batch conversion tool was essential for this research. The laszip tool functioned perfectly fine, however, we discovered utilizing this tool condensed the conversion allowing us to convert all forty LAZ files. Using the batch conversion alternative allowed our group to convert multiple files once for greater efficiency.

The training model required the CPCUTM analysis tool. We followed the instructions based off the tutorial on ArcGIS Online. However, we discovered that when we entered our Input Model Definition accordingly and browsed to the file indicated, our training model was incorrect. The training model was unable to classify any transmission poles and towers within the training data. We detected the error and altered the Input Model Definition to an alternative folder within the Results folder to small data checkpoints rather than large. We also noticed the epoch number when selecting the file was not the same for all group members. For one of our members, the number did not align with the tutorial. Choosing the epoch accordingly for each of our highest recall number solved the issue and allowed our group members to successfully classify transmission poles and towers correctly.

Another issue our group encountered was the inability to classify transmission towers and poles despite being given the correct classification codes. The training model was successful in this identification, but when we incorporated SCE’s data, we noticed there was no transmission towers and poles being correctly classified. We were able to locate them through thorough inspection of our results, but the color did not correspond to the classification code. This led our group to retrain the model several times. One group member retrained the model multiple times as their computing hardware was able to process the model at a much faster rate. After several cycles, we concluded that our model was unsuccessful in classifying transmission towers and poles.

Running the CPCUTM geoprocessing tool took an extensive amount of time. One of our group member’s CPCUTM took a prolonged duration the first time it was trained, such as 5 hours. However, once the model was rerun to be trained again it was reduced to 4 hours. Another team member only took roughly 30 minutes to complete. Depending on the device, the CPCUTM can take much longer than anticipated. Differing levels of computer hardware and processing abilities among the members of the group was a challenge that the group had to overcome. For instance, ArcGIS Pro needs to be installed for all users or there will be a concern pertaining to permissions. The differences in computing power were an obstacle that the group had to conquer due to the project’s deadline.

**Discussion and next steps**

Southern California Edison must rely on third party vendors to process LiDAR data obtained for numerous purposes, involving the identification of assets in the service territory, and improving the geolocation data of these assets. The process of examining service assets requires a substantial amount of labor, high expenses, and is challenging to execute. A LiDAR data-processing tool developed for SCE provides the possibility to remove the need for expensive vendors and allows SCE better control and enhanced quality of assets’ geolocation data. Lidar is an optical remote-sensing method that uses laser light to densely sample the surface of the earth and is a cost-effective alternative to conventional surveying procedures. Lidar produces point-cloud datasets that can be managed, visualized, analyzed, and shared using software, such as ArcGIS Pro.

Teams can use a collection vehicle, laser scanner system, GPS (Global Positioning System), and INS (inertial navigation system) to scan large areas and then analyze the data to identify which assets require attention. This focuses efforts on assets needing consideration. This study desired to demonstrate the feasibility of a LiDAR point cloud data-processing tool and to study its possible benefits. The literature verified the successful application of deep learning to correctly classify some objects. Thus, this study sought the use of a deep-learning model to justify whether this was a feasible method.

Therefore, ArcGIS Pro is equipped with tools to collaboratively classify points. These tools can be used to identify points of interest in the training and service territory data. This data should be a precise depiction of the assets of interest. Then, a model should be trained using the training and service territory data. These actions should generate a model that can be implemented to automatically classify other point-cloud datasets.

**Conclusions**

This document focuses on machine learning to identify assets in the service territory from point-cloud data. The literature suggests deep learning can be used as a tool to classify objects of interest. This study sought to demonstrate the use of a deep-learning model to determine whether deep learning is a viable solution for identifying high vegetation, transmission poles, and transmission towers. This study created a training model from a tutorial in ArcGIS Online to determine whether deep learning is an effective solution for extracting the desired objects from LiDAR data. The deep-learning model successfully identified high vegetation and transmission wires in most of the SCE data. However, the model was unable to classify transmission poles correctly and only identified roughly half a transmission tower. This study supports the literature that deep learning can successfully classify point clouds to some extent.

## **References**

Antunes J. (2018). Should You Choose LiDAR or Photogrammetry for Aerial Drone Surveys? Commercial UAV News. On the WWW, February. URL <https://​www​.commercialuavnews​.com​/construction​/choose​lidar​photogrammetry​aerial​drone​surveys>.

Azevedo F, Dias A, Almeida J, Oliveira A, Ferreira A, Santos T, Martins A, and Silva, E. (2019). LiDAR-Based Real-Time Detection and Modeling of Power Lines for Unmanned Aerial Vehicles. MDPI. On the WWW, April. URL <https://​doi​.org​/10​.3390​/s19081812>.

Borcs A, Balazs N, and Benedek C. (2017). Instant Object Detection in LiDAR Point Clouds. Digital Object Identifier System. On the WWW, March. URL <https://​doi​.org​/10​.1109​/LGRS​.2017​.2674799>.

Brubaker K M, Myers W L, Drohan P J, Miller D A, and Boyer E W. (2013). The Use of LiDAR Terrain Data in Characterizing Surface Roughness and Microtopography. Applied and Environmental Soil Science, 2013, Article 891534. See also URL [https://​doi​.org​/10​.1155 ​/2013​/891534](https://​doi​.org​/10​.1155%20​/2013​/891534).

Fan Y-C, Yelamandala C M, Chen T-W, and Huang C-J. (2021). Real-Time Object Detection for LiDAR Based on LS-R-YOLOv4 Neural Network. Journal of Sensors, 2021, Article 5576262. See also URL <https://​doi​.org​/10​.1155​/2021​/5576262>.

Glavinich T E, and Chichester A L. (2021). Pricing Service Work for Profit. Electrical Contractor Magazine. On the WWW, July. URL <https://​www​.ecmag​.com​/section​/your​business​/pricing​service​work​profit>.

Kudinov D. (2019). PointCNN: Replacing 50,000 Man Hours with AI. Medium. On the WWW, July. URL <https://​medium​.com​/geoai​/pointcnn​replacing​50​000​man​hours​with​ai​d7397c1e7ffe>.

Maggiori E, Tarabaka Y, Charpiat G, and Alliez P. (2017). Convolutional Neural Networks for Large-Scale Remote-Sensing Image Classification. On the WWW, February. URL <https://​doi​.org​/10​.1109​/TGRS​.2016​.2612821>.

Nahhas F H, Shafri H Z M, Sameen M I, Pradhan B, and Mansor, S. (2018). Deep Learning Approach for Building Detection Using LiDAR–Orthophoto Fusion. Journal of Sensors, 2018, Article 7212307. See also URL <https://​doi​.org​/10​.1155​/2018​/7212307>.

National Ocean Service. (2021). What Is LiDAR. US Department of Commerce. On the WWW, December. URL <https://​oceanservice​.noaa​.gov​/facts​/lidar​.html>.

Prokhorov D V. (2009). Object Recognition in 3D LiDAR Data with Recurrent Neural Network. IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pp. 9–15. See also URL <https://​doi​.org​/10​.1109​/CVPRW​.2009​.5204114>.

Singh K K, Chen G, McCarter J B, and Meentemeyer R K. (2015). Effects of LiDAR Point Density and Landscape Context on Estimates of Urban Forest Biomass. ISPRS Journal of Photogrammetry and Remote Sensing, 101, 310–322. See also URL <https://​doi​.org​/10​.1016​/j​.isprsjprs​.2014​.12​.021>.

Sultan, V., Ramirez, J., Peabody, J., & Bautista, M. (2022). Managing power infrastructure using LiDAR.

Van Leeuwen M, and Nieuwenhuis M. (2010). Retrieval of Forest Structural Parameters Using LiDAR Remote Sensing. European Journal of Forest Research, 129: 749–770. See also URL <https://​doi​.org​/10​.1007​/s10342​010​0381​4>.

Van Tassel C. (2021). The True Cost—Implementing LiDAR into Your Business. LiDAR News. On the WWW, February. URL <https://​lidarnews​.com​/articles​/the​true​cost​implementing​lidar​into​your​business>.