Journal of High School Science


# SurfaceNet: leveraging aerial LIDAR point clouds for post-earthquake building damage assessment

Fan C

Submitted: July 17, 2024, Revised: version 1, September 14, 2024
Accepted: September 16, 2024

**Abstract**
Rapid mapping of post-earthquake damage is important for effectively staging rescue efforts, given that 90% of victims do not survive more than three days when trapped. Consequently, computational approaches to damage mapping have been explored to improve speed. Automatic building damage detection methods can produce results faster than human-based assessments, which has made them increasingly popular in both research and real-world applications. Applications of aerial LIDAR data in particular show high potential in this regard, because aerial LIDAR data can be obtained relatively quickly and contains valuable geometric information, which is inherent to patterns of collapsed buildings. Deep learning approaches have been recently explored for assessing the damage to buildings. However, most general point cloud deep networks have been designed using autonomous driving datasets and multi-view single-object datasets in mind, which negatively impacts their performance on assessing the damage to buildings from aerial LIDAR data. Consequently, we aimed to design an architecture embedding prior domain knowledge of aerial LIDAR for building damage assessment. We propose SurfaceConv, a novel local aggregator module, which enables consideration of long-ranged intra-surface dependencies and thus, a holistic consideration of distinct surfaces. Using the SurfaceConv module, we created SurfaceNet, a deep neural network (DNN) specifically tailored for detecting damage to buildings using aerial LIDAR data. We achieved results in the previously explored patch segmentation task that were comparable to existing models. We introduced the classification and general segmentation task and found that our model empirically outperformed all existing models. These existing models that we benchmarked against included both generalized DNNs trained from scratch, generalized pretrained DNNs, and building damage assessment specific DNNs.

**Keywords**
Post earthquake building damage, Collapsed building, Earthquake, Aerial LIDAR, 3D point clouds, 3D deep learning, Deep Neural Network, Point cloud model, Local aggregator, Segmentation method

Cory Fan, Shanghai American School, Pudong Campus, 1600 Lingbai Road, Pudong District, Shanghai, Shanghai, China 201201. Cory01pd2025@saschina.org

## 1. Introduction

In the hours, days, and weeks after earthquakes, rapid and accurate assessments of damage serve as a critical part of recovery from natural disasters. Detecting collapsed and heavily damaged buildings is important for effectively assigning search and rescue teams and minimizing loss of lives, allowing teams to prioritize resource allocations to more damaged regions where residents are likely to be in more need of help (1).

Building damage assessment can play a substantial role in post-earthquake rescue. INSARAG (the International Search and Rescue Advisory Group) guidelines outline five major steps to a recovery operation: "Wide Area Assessment", "Worksite Triage Assessment", "Rapid Search and Rescue", "Full Search and Rescue", and "Total Coverage Search and Recovery". Building damage assessment, when possible, is involved in parts of the "Wide Area Assessment" step, which involves assessing the overall damage of sectors and parts of the "Worksite Triage Assessment", which involves determining and assessing specific rescue sites within sectors (2). Thus, building damage assessment is crucial for efficient allocation of rescue teams (3).

The rapidity of this building damage assessment is critical. While the survival rate of a trapped victim is around 90% within the first twenty-four hours, that statistic drops to < 10% after the first three days (4). Consequently, the rapidity of any building damage assessment is crucial for search and rescue purposes.

Traditionally, building damage assessments have been performed through field surveys, with human investigators examining buildings on the ground; however, field surveys are too slow to aid decision-making in the days following the earthquake. Predictive software, such as PAGER https://earthquake.usgs.gov/data/pager/ (Prompt Assessment of Global Earthquakes for Response) has been previously utilized for allocation of post-earthquake rescue resources. However, this software is meant primarily to provides fatality and economic loss impact estimates following significant earthquakes worldwide. It does not predict at-site building damage (5). For actual and rapid damage assessment, interpretation of remote-sensing data, which can cover the full area of the earthquake and be acquired quickly, has been the method of choice. For example, rapid damage assessments are often performed by organizations such as UNOSAT (The United Nations Satellite Center) or Copernicus EMS through human visual interpretation of satellite imagery (6). More recently, Deep Neural Networks (DNNs) have been utilized as well, such as the usage of CNNs (Convolutional Neural Networks) on satellite imagery (7, 8).

3D point clouds, typically sourced from aerial LIDAR (LIght Detection And Ranging), have been considered for building damage assessment (9, 10). This aerial LIDAR is generally collected by an unmanned aerial vehicle. Unlike satellite images, aerial LIDAR contains height information, which is especially useful for building damage assessment, since perturbations in height, or other abnormal geometric structures, can be indications of damage. The geometric information of LIDAR may capture damage that images cannot – for instance, an abnormal

tilting of a roof-surface would not change the color of the roof on an image, or necessarily have any obvious damage artifacts, but would be easy to discover in a 3D point cloud. Moreover, compared to images or the human eye, LIDAR does not require light and can work in the dark, which can be beneficial for rapid responses to earthquakes that occur at night. To process this LIDAR data, some recent papers have proposed using point network deep learning (9-12). Some methods (12) utilize pre- and post-earthquake LIDAR data; however, since collecting LIDAR is an active process, we choose to focus on methods that only require post-earthquake LIDAR data.

Traditionally, point cloud deep learning has been driven by multi-view-generated point cloud datasets (such as ModelNet40, ShapeNetPart, or ScanObjectNN) and single-view 360°-point cloud datasets for autonomous vehicles (such as SemanticKitti or NuScenes). Papers exploring deep learning architectures for aerial LIDAR point clouds have been comparatively few. In the domain of building damage assessment, one deep learning architecture is DS-Net, which is motivated by the characteristics of damaged and undamaged buildings (10). We similarly believe that designing deep network architectures motivated by prior observations of the characteristics of buildings in aerial LIDAR point clouds can improve functionality; consequently, we propose a novel deep learning architecture, SurfaceNet, which takes advantage not only of the characteristics of damaged buildings but also the characteristics of aerial LIDAR.

The key contributions of this paper are summarized as 1] We propose SurfaceNet, a deep learning model which targets aerial-LIDAR building damage assessment, which utilizes our novel SurfaceConv module. 2] We propose the local aggregator module, SurfaceConv, which utilizes contiguous receptive fields to propagate information within individual surfaces. We explain the motivation in 4.1. 3] SurfaceNet exceeds existing SOTA point cloud methods by 2.2% accuracy in classification, 1.1% mIOU in patch segmentation (defined in Section 2.1.2) and 0.5% mIOU in general segmentation on a dataset constructed from the 2016 Kumamoto Earthquake. 4] Our model additionally outperforms existing methods by 0.6% mIOU on general segmentation on a dataset constructed from the 2010 Haiti Earthquake, indicating the success of our model on a variety of building types.

## 2. Materials and Methods

### **2.1** Dataset

We trained our model on two datasets, one created with data from the 2016 Kumamoto Earthquake (which we will refer to as KumamotoEQ), and another from the 2010 Haiti Earthquake (which we will refer to as HaitiEQ). We utilized KumamotoEQ as our primary evaluation dataset, and HaitiEQ as our secondary evaluation dataset. We utilize HaitiEQ to demonstrate that our model could succeed on datasets with a variety of earthquake intensities or building types, not just the Japanese buildings presented in KumamotoEQ. We chose KumamotoEQ as our primary evaluation dataset due to its high-quality building damage labels originating from field surveys (13), and the presence of a building footprint map (14). On the other hand, HaitiEQ has less reliable labels (15), such that we could not identify building footprint maps;

hence, we utilized it as our secondary evaluation dataset.

2.1.1 *KumamotoEQ study area and description*

For the KumamotoEQ dataset, our study area was centered around the Mashiki suburb of Kumamoto (shown in Figure 1), which experienced the greatest effects of the earthquake and consequently included the greatest concentration of damaged buildings. We utilized only the post-earthquake LIDAR data, which was provided by Air Survey Co., Ltd., of Japan to OpenTopography (16), as well as building footprints provided by the Geospatial Institute of Japan (14) and damage assessments from the Building Research Institute of Japan and Architectural Institute of Japan Kyushu, as described by Yamazaki et al. (13). The LIDAR data was captured using a Leica ALS5011 Laser Scanner mounted to a Cessna 208 aircraft. The collection altitude was ~ 2000-2100 meters above ground, and the pulse rate was 110 kHz (17). The dataset contains 2388 buildings, which are primarily constructed of wood (18). The number of buildings in the dataset is slightly more than the total number of labeled buildings due to instances where two buildings were grouped together in the damage labeling but split in the building footprint.

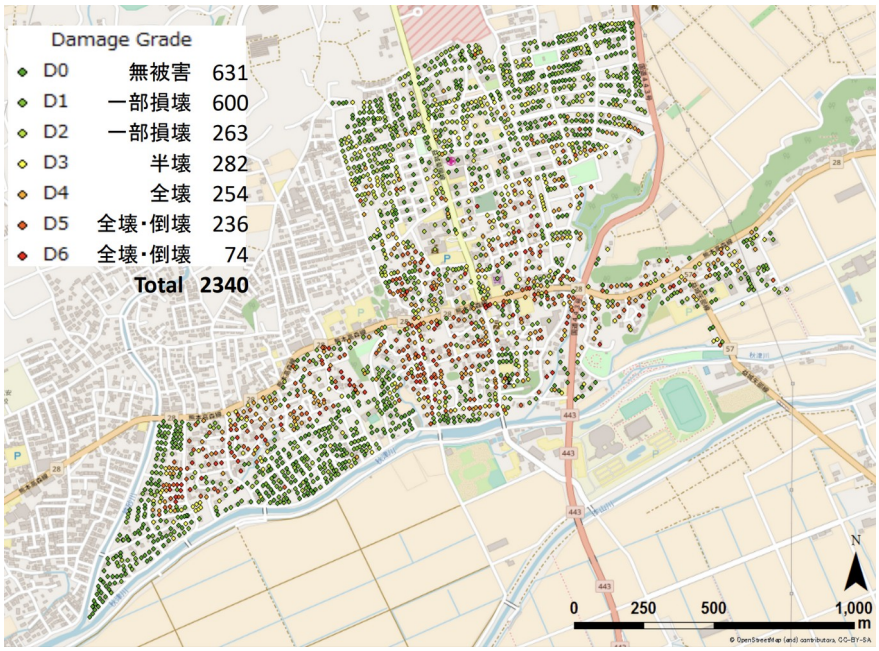


**Figure 1.** The KumamotoEQ Study Area (13)

The labels provided by the Geospatial Institute of Japan followed the building damage system established by Okada and Takai (19). Following prior convention (10), we designated D0-D4 as uncollapsed buildings and D5-D6 as those that had collapsed. Under this convention, the dataset contained 2071 uncollapsed buildings and 317 collapsed buildings.

2.1.2 *HaitiEQ study area and description*

For the HaitiEQ dataset, our study area was a rectangular area located in Port-au-Prince, bounded ~ by latitudes 18.5407 and 18.5558

and longitudes -72.3493 and -72.3290 (20). We utilized the post-earthquake LIDAR data jointly collected by the Rochester Institute of Technology and Kucera International, which was collected with a Leica ALS60 with pulse rate 150 kHz at ~ 820 meters altitude (15). The data is publicly hosted on OpenTopography (21). The labels we utilized were from the HaitiBRD dataset, produced by researchers from the National Taiwan University and Asia University. They consisted of a large geolocated image containing segmentation mask labels for each individual pixel. The dataset contained over 9000 buildings (20); however, the dataset did not provide individual building footprints. We matched points to pixels to label each individual point. Most buildings in the Haiti dataset were concrete or earth (22).

Since the labeling schema of the HaitiBRD damage masks was unknown (23), we split "No Damage" and "Minor Damage" classes into the uncollapsed class, and the "Major Damage" and "Destroyed" classes into the collapsed class. Under this convention, since the dataset did not differentiate between individual buildings, the dataset contained 6,550,317 non-building (background) points, 5,675,991 uncollapsed building points, and 697,109 collapsed building points.

2.1.3 *Data preparation*

We performed three separate tasks to evaluate our model: classification, patch segmentation, and general segmentation. For the KumamotoEQ dataset, we trained and evaluated all models on all three tasks. For the HaitiEQ dataset, due to the lack of building footprint maps, we only trained and evaluated models on the general segmentation dataset.

"Patch" segmentation refers to the task of performing segmentation on individual building "patches" -- point clouds defined by bounding boxes surrounding the building with some padding. While not true segmentation, we performed patch segmentation for comparability with Xiu et al.'s (10) method. "General" segmentation refers to the more common segmentation data preparation method of splitting a larger point cloud area into individual tiles (containing multiple, one, or no buildings) and performing segmentation on that tile.

For classification and patch segmentation, we produced building patches by taking the latitude-longitude bounding box of each building footprint. Following Xiu et al., (10), we applied padding to the bounding boxes. We utilize $1x10^{-4}$ padding for both longitude and latitude, which equates to ~ 11 meters of padding for latitude and ~ 9 meters for longitude. This allowed the model to also consider the surroundings of the building, which could contain context such as rubble.

For the purpose of general segmentation, we split the entire area into 100 x 100 $m^2$ point clouds, which was ~ 32,000 points per tile. This number was chosen because it was the maximum integer that would fit into the memory of an RTX 4090 (NVIDIA® GeForceRTX™4090 GPU). A single batch of 200 x 200 $m^2$, for example, on the DS-Net will not fit into the memory of an RTX 4090).

We split each dataset into train, validation, and test according to a 70-10-20 ratio respectively. For classification and patch segmentation, we used a stratified split across each individual damage class, as opposed to a stratified split

only on damaged and undamaged classes. This resulted in two separate datasets, our KumamotoEQ and HaitiEQ datasets, which we used to evaluate our models.

**2.2** SurfaceNet

In this section, the design and motivation of/for the SurfaceNet architecture are discussed. First, we analyze the motivating characteristics of aerial point clouds of buildings with respect to SurfaceConv. Next, we discuss the actual design of SurfaceConv. Then, we present the SurfaceNet architecture itself. Finally, we discuss tasks and training hyperparameters.

2.2.1 *Motivations*

Inductive bias has played an extremely significant role in the development of deep neural networks (DNNs). Inductive bias refers to the assumptions that make a model more likely to learn patterns of a certain nature. Before the advent of deep neural networks, inductive bias would often be specifically defined – for instance, the KNN classification algorithm assumes that similar points in a feature-space will tend to have similar labels. In DNNs, one instance of inductive bias is the convolution operator that forms the basis of CNNs. The convolution operator enforces an emphasis on local information through limiting the receptive field, since pixels that are near to each other are more likely to have dependencies that are important to model. In generalized deep point cloud models, inductive bias towards modeling local dependencies is usually used, through "local aggregator" modules. These modules usually aggregate information from a spherical local neighborhood, using techniques such as graph convolution.

Within the confines of the application of deep point cloud models to building damage assessment , the design of inductive bias has been previously investigated. Prior papers have introduced inductive bias towards the properties of buildings regardless of point cloud source; for instance, DS-Net's proposal emphasized local smoothness (approximated *via* the Discrete Laplacian) as a differentiating factor between collapsed and non-collapsed buildings (10). However, DS-Net does not consider characteristics of aerial LIDAR. Other papers have proposed inductive bias towards the characteristics of aerial LIDAR. Nong et al. (24) proposed a segmentation upsampling method which placed a special emphasis on the relative elevations of points, but maintained the local, radial receptive field of the original PointNet$^{++}$ (25). On the other hand, no existing deep-learning methods have considered a unified approach containing both prior observations of the characteristics of building damage assessment and prior observations of the characteristics of aerial LIDAR data, although some methods have considered the two components independently.

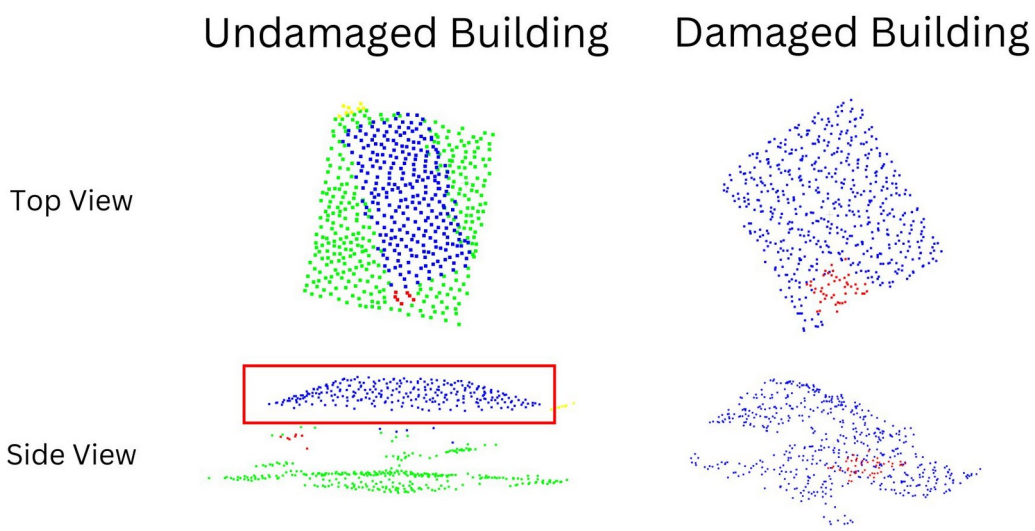


**Figure 2.** Examples of Undamaged and Damaged Buildings. Distinct colors are surfaces found by the algorithm described in Section 2.2.2. Notice that the undamaged building contains a distinct roof surface (blue), while the damaged building is largely a single, irregular surface.

We propose an inductive bias, through our SurfaceConv module, which leverages general observations regarding the overall characteristics of building damage assessment with the aerial LIDAR format. We made two observations that motivate our inductive bias. First, we observed that aerial LIDAR scans of undamaged buildings often resulted in a few distinct disconnected roofs and ground surfaces, while damaged buildings tended to result in point clouds without distinctive regular surfaces (see Figure 2 for a visual explanation). Second, given the first observation, for the task of building damage assessment from aerial LIDAR, we observed that points on the same surface tended to have similar semantic meaning, such as being a part of a building roof, or the top of a tree, or part of a car roof. Given these two observations, we believed that holistic consideration of these surfaces was important for understanding their semantic meaning, such as whether or not the surface was part of a damaged or undamaged building. As such, we argue that, for building damage assessment models, an inductive bias that emphasizes intra-surface relationships is necessary.

Indeed, previous non-machine-learning methods for building damage assessment have emphasized the holistic consideration of distinct surfaces. For instance, Axel and Aardt (26) utilized a region-growing approach using a non-morphological filter to aggregate points on the same surface before considering whether these points constituted a damaged or undamaged surface as a whole. We provide a discussion our definition of surface for point clouds in Section 2.2.2.

This inductive bias is not present in generalized point cloud architectures. Generalized point cloud architectures use a hierarchical scheme of composing local information gradually into a global whole, similar to CNNs (26-28). This constitutes an inductive bias towards emphasizing local relationships, while disregarding the surface-relationships. However, because of the long distance between one end of a surface and another, these networks may consider intra-surface relationships in the same aggregative stage as inter-surface relationships, which does not take into consideration the special relationship that points on the same surface have. The core motivation of this paper was to change the hierarchical aggregative scheme to compose local information into intra-surface information, before finally composing the information to a global understanding of the point cloud.

2.2.2 *SurfaceConv*

Since point clouds are inherently geometric, most models that operate on point sets utilize locally biased aggregators of some form. Broadly, most local aggregators take on the form

$$f(x_i) = h(\{ g(x_i, x_j, p_i - p_j) \mid A_{ij} = 1\}) \quad (1)$$

where $g$ is a learned function (e.g. MultiLayer Perceptron (MLP) or attention), $h$ is a set aggregation function (e.g. max-pooling or summation), $x$ is the feature vector, $p$ is the position vector, and $A$ is the adjacency matrix created from some sort of local relationship modeling (e.g. KNN or Ball-Query), with self-loops. In contrast to the radial receptive fields created by KNNs (K-Nearest-Neighbor) or Ball-Queries, SurfaceConv leverages a surface-aware receptive field.

Two points can be identified as being on the same surface if there exists an ordering of points that form a path between these two points given that the distance between two consecutive points on the said path are sufficiently small. Axel and Aardt, 2017 (26) implemented this idea in their algorithm with region growing, with the additional constraint that surfaces should be relatively flat. Another implementation of this idea is to construct an undirected k-nearest-neighbor graph and define two points as being on the same surface if they are in the same connected component. However, due to constraints on the shape of tensors, efficient implementation of the max-pool aggregator requires the set of neighbors to be of equal cardinality for all points; moreover, connected components can be very large, such as ground surfaces, wherein two points that are very far from each other are not relevant to each other. Consequently, we approximated the implementation, and let the immediate receptive field of SurfaceConv (the points from which SurfaceConv aggregates information) be all points reachable by N-hops on a directed KNN graph. This could be implemented efficiently with an iterative approach. Formally, letting the function $f'$ represent the SurfaceConv operation

$$f^{(m)}(x_i) = Max(\{g(x_i, x_j, p_i - p_j) \mid A^m_{ij} > 0\}) \quad (2)$$

$$f'(x_i) = h(f^{(1)}(x_i), \ldots f^{(M)}(x_i)) + x_i \quad (3)$$

where $g$ is a linear transformation, $h$ is a linear transformation followed by a normalization operation, $x_i \in \mathbb{R}^{N \cdot C}$ is the feature matrix, $p_i \in \mathbb{R}^{N \cdot 3}$ is the position matrix and $A$ is the directed adjacency matrix generated by a KNN operation with self-loops. The usage of $g$ as a linear transformation was motivated by computational efficiency, since linear transformations are covered by the distributive property. By taking the $m^{th}$ power of $A$, we could generate a matrix of the n-hop neighborhood surrounding each node. Increasing M increased the size of the receptive field; decreasing M correspondingly decreased the size of the receptive field. We found M=3 to be a sweet spot. Increasing K also increased the size of the receptive field but could cause the model to "jump" over gaps across surfaces. On the other hand, due to the natural stochasticity in the position of points in a point cloud, setting K to a value that is too small resulted in the receptive field having a high degree of noise based off slight changes in the position of points. We found K=8 to be an approximate sweet spot. We included a Batch Normalization operation before the residual connection for stability. The computation of SurfaceConv is presented visually in Figure 3.

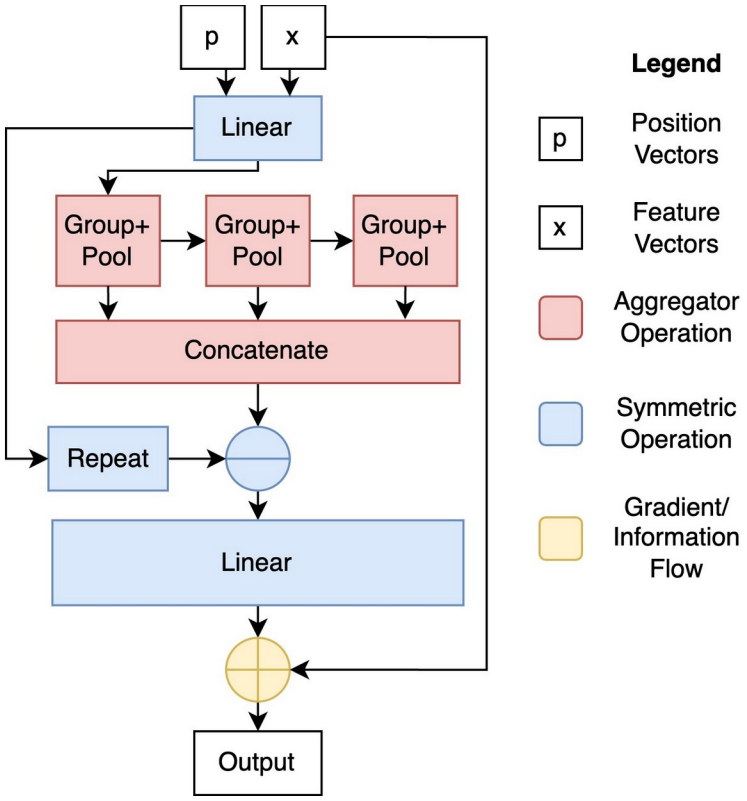


**Figure 3.** Surfaceconv computation flowchart

### 2.2.3 *SurfaceNet architecture*

We followed the conventional hierarchical max-pooling residual MLP structure for point cloud models.

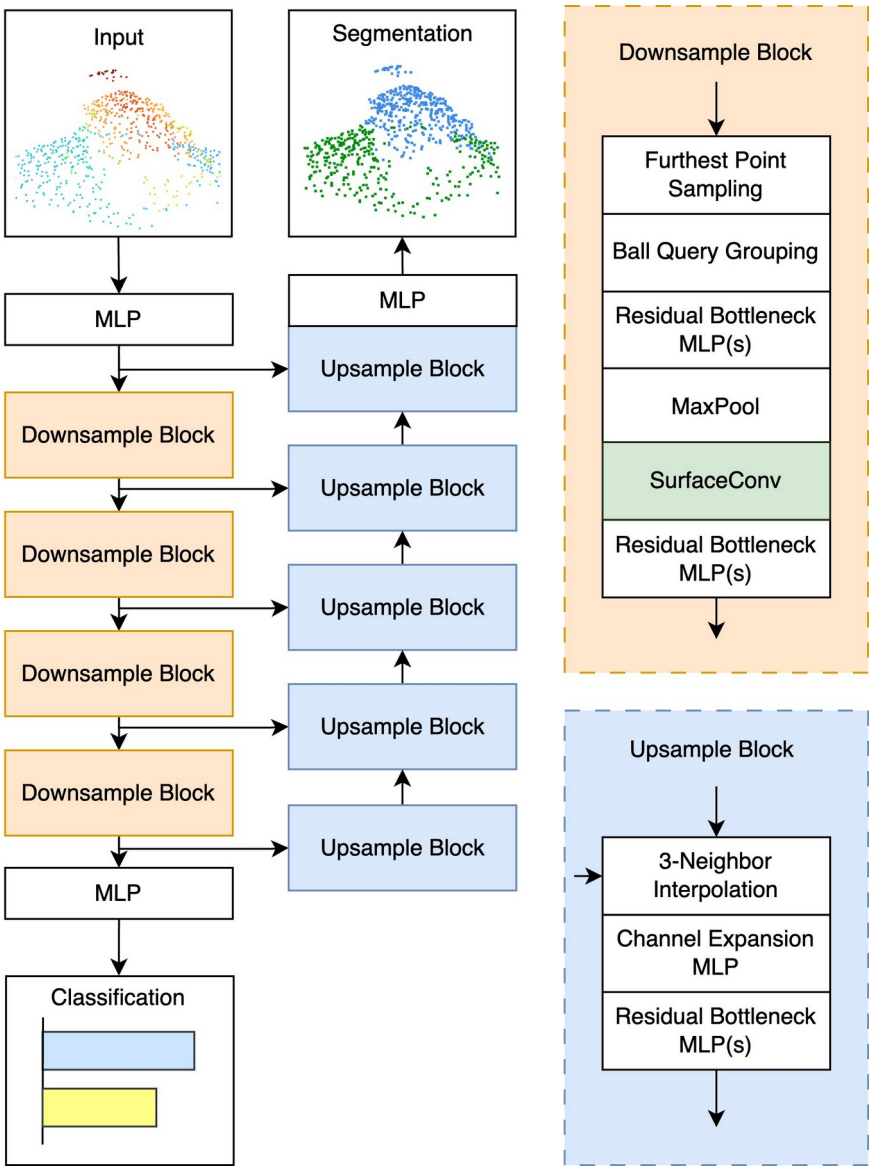


**Figure 4.** SurfaceNet computation and flowchart

For the encoder, we utilized both SurfaceConv and the conventional local aggregator detailed in PointNet$^{++}$ (25), which utilizes Query-Ball Grouping followed by a pointwise MLP and max-pooling with farthest-point-sampling down-sampling. We modeled local and surface-external relationships using this conventional local aggregator, while we modeled surface-internal relationships using SurfaceConv. For the segmentation decoder, we utilized a U-Net structure with the 3-Nearest-Neighbor interpolation proposed by

PointNet$^{++}$. Additionally, since segmentation is a more complex task than classification, we expanded the number of input channels from 32 to 64 and reduce the bottleneck from a factor of 4 to a factor of 2. Additionally, we varied the downsampling rate for the three tasks: the classification encoder downsampled by a factor of [1, 2, 2, 2] for the four stages respectively, the patch segmentation encoder downsampled by a factor of [2, 4, 4, 4], and the general segmentation encoder downsampled by a factor of [8, 4, 4, 4]. Incorporating a greater degree of down-sampling for segmentation was inspired by PointMLP (27). Additionally, for both the segmentation tasks, the relative xyz-positions of each point were concatenated to the features of each point in the "sample $^{+}$ group" stage, which embedded further positional emphasis which is highly relevant for segmentation tasks. Channel expansion between each down-sampling layer remained at a factor of 2 for both segmentation and classification.

**2.3** Other tested models

We compare our model against several existing deep networks. These models consisted of four fully supervised 3D point cloud networks, one pretrained 3D point cloud network, and one building-damage-specific point cloud network. The four fully supervised networks and the pretrained networks were designed primarily with multiview object point cloud datasets in mind, such as the ModelNet40 and ScanObjectNN.

Of the four fully supervised methods, three are based off of the traditional max-pooled aggregator: PointNet (29), PointNet$^{++}$ (26), and PointMLP (27). PointNet is one of the seminal models of 3D point cloud networks, establishing the formulation of the max-pool aggregator discussed in Equation 1 (section 2.2.2). It utilized two aggregators with global receptive fields. PointNet$^{++}$ is the extension of PointNet's aggregator to a hierarchical structure similar to CNNs, with local, not global, aggregators and downsampling. Finally, PointMLP is a more recent formulation of the max-pool local aggregator, incorporating residual connections and is an overall deeper and larger model.

The remaining fully supervised network was DGCNN (30), or "Dynamic Graph CNN". It utilizes the max-pool aggregator with a dynamic receptive field, by drawing the adjacency matrix using distance in the feature space rather than in the XYZ space. In this way, DGCNN can dynamically learn the relationships it seeks to model.

The pretrained model was a PointMLP checkpoint pretrained with ULIP (31). ULIP performs multimodal alignment by bootstrapping the point cloud modality with a more powerful pre-aligned vision-language model, specifically CLIP. ULIP produces a pretrained point cloud encoder which can be applied to classification tasks. We tested with ULIP to determine whether knowledge transfer from object-oriented datasets, such as ShapeNet, could significantly improve performance on point clouds with aerial LIDAR.

Finally, the building-damage-specific point cloud network tested against was DS-Net Large, the large variant of DS-Net. This network leverages the Discrete Laplacian as a local aggregator, arguing that the usage of the Discrete Laplacian can measure the smoothness

of surfaces and thus measure characteristics of buildings that are specifically useful for assessing the damage to buildings.

**2.4** Training methods

As specified in section 2.1.2, we trained on three separate tasks: classification, patch segmentation, and general segmentation. All fully-supervised models specified in section 2.3 were trained on the KumamotoEQ dataset and the HaitiEQ dataset for their respective tasks. For pretrained models specified in section 2.3, we used open-source checkpoints and subsequently finetuned on the KumamotoEQ and HaitiEQ datasets. We trained with AdamW optimizer and Cosine Annealing Scheduler. We utilized random isotropic scaling for data augmentation. We trained with inverse class frequency weighted cross entropy to address the class imbalance. Although stratified split ensured equivalent class-distributions between training, validation, and testing, the usage of standard, non-weighted cross entropy, in our experiments, caused the model to converge to only predict the most represented class; consequently, inverse class frequency weighted cross entropy was necessary to improve training speed and overall trainability.

Hyperparameters were tuned only on the validation performance. For testing, we selected the best performing checkpoint under validation and assessed it under the test set. Specific hyperparameters across the three tasks can be seen in Tables 1 and 2.

**Table 1.** From-scratch training hyperparameters.

| Hyperparameter | Classification | Patch Segmentation | General Segmentation |
|---|---|---|---|
| Learning Rate | 0.1 | 0.001 | 0.0001 |
| Weight Decay | $2x10^{-5}$ | 0 | 0 |
| Epochs | 600 | 1000 | 500 |
| Number of Points | 600 | 600 | 8000 |
| Batch Size | 32 | 32 | 3 |
| Checkpoint Metric | Class Accuracy | Mean IOU | Mean IOU |

**Table 2.** Fine-tuning the hyperparameters

| Hyperparameter | Classification |
|---|---|
| Learning Rate | 0.01 |
| Weight Decay | 0 |
| Epochs | 600 |
| Batch Size | 32 |

For calculating class accuracy, a building was predicted as damaged if it was assigned > 50% probability for the damaged class, and undamaged otherwise. For calculating mean IOU, we considered a model, from the three classes (including both the two foreground and the background class), to have predicted a class if its corresponding probability was the highest

among the three classes. This was most analogous to a threshold of 0.5 on a binary segmentation task.

## 3. Results

We compared the results of SurfaceNet with several mainstream open-source point cloud models, along with some existing non-deep learning algorithmic methods. We re-implemented DS-Net; which was inadequate due to lack of detail in its network structure. However, we corrected this discrepancy by filling in missing details based on background knowledge of such normal point cloud network designs.

### 3.1 KumamotoEQ

As shown in Table 3, on the KumamotoEQ dataset, SurfaceNet outperformed all existing methods for building damage assessment from aerial LIDAR, including generalized point cloud DNNs trained from scratch, generalized pre-trained point cloud DNNs, and building-damage-specific-specific DNNs. Specifically, SurfaceNet outperformed existing models by 2.2% class accuracy on classification, 1.1% mIOU on patch segmentation, and 0.5% mIOU on general segmentation.

**Table 3.** KumamotoEQ experiment results. Best results are bolded, second-best results are underlined. Blank spaces (denoted by "-") indicates that there is no open implementation (or checkpoint for pretrained models) for that model on that given task. Green bold text indicates improvement over the next-best model.

| Method Type | Method | Classification (Class Accuracy) | Patch Segmentation (Mean IOU) | General Segmentation (Mean IOU) |
|---|---|---|---|---|
| Fully-Supervised DNN | PointNet (29) | 78.6 | - | - |
| Fully-Supervised DNN | PointNet++ (26) | 90.6 | 49.6 | 39.5 |
| Fully-Supervised DNN | DGCNN (30) | 86.0 | - | - |
| Fully-Supervised DNN | PointMLP (27) | 86.0 | 50.1 | 41.1 |
| Pre-Trained DNN | ULIP + PointMLP (31) | 88.4 | - | - |
| Building-Damage-Specific-DNN | DS-Net Large (10) | - | 49.6 | 42.3 |
| **Proposed (Fully-Supervised)** | **SurfaceNet (this work)** | **92.8** (+2.2) | **51.2** (+1.1) | **42.8** (+0.5) |

### 3.2 HaitiEQ

We additionally performed experiments on the HaitiEQ dataset to explore the generalizability of our model, as mentioned in section 2.1.1. As shown in Table 4, SurfaceNet outperformed all tested models, demonstrating an improvement in the performance of 0.6% mIOU when compared with the next-best model. Notably, DS-Net Large failed to converge on the dataset, achieving a 25.8% mIOU. This indicateed that our model could predict damage to buildings regardless of construction type or materials such as wooden Japanese buildings and concrete or earth Haitian buildings. However, these results should be used primarily comparatively between various models; the

absolute mIOU values may not be particularly accurate. This is due to the lack of validation of the labeling quality (as discussed in section 2.1.1).

**Table 4.** HaitiEQ experiment results. Best results are bolded, second-best results are underlined. Green bold text indicates improvement over the next-best model.

| Method Type | Method | General Segmentation (Mean IOU) |
|---|---|---|
| Fully-Supervised DNN | PointNet++ (26) | 37.8 |
| Fully-Supervised DNN | PointMLP (27) | 45.0 |
| Building-Damage-Specific-DNN | DS-Net Large (10) | 25.8 |
| **Proposed (Fully-Supervised)** | **SurfaceNet (this work)** | **45.6 (+0.6)** |

**3.3** Ablation studies

We performed ablation testing on the general segmentation task of the KumamotoEQ dataset, in order to understand the effect of varying the hyperparameters of the SurfaceConv model, our key contribution. Varying the values of M and K resulted in minor decreases in performance; however, the performance for all except the M=4 test resulted in equal or greater performance than the next-best model, DS-Net Large (Table 5). Additionally, as indicated by the HaitiEQ set of experiments, DS-Net Large itself has unidentified, deficiencies. All versions of SurfaceNet outperformed PointMLP. Consequently, SurfaceNet was not overly sensitive to the choice of M and K.

**Table 5.** Ablation experiment results on KumamotoEQ general segmentation.

| M-Value | K-Value | General Segmentation Performance (Mean IOU) |
|---|---|---|
| 3 | 6 | 42.6 (-0.2) |
| 3 | 10 | 42.3 (-0.5) |
| 2 | 8 | 42.4 (-0.4) |
| 4 | 8 | 41.5 (-1.3) |
| 3 | 8 | 42.8 (Original) |

## 4. Discussion

**4.1** Analysis of results

The success of SurfaceNet against generalized methods demonstrateed the value of targeted architectures for building damage assessment. Due to their nature as generalized point cloud models, they do not specifically contain inductive bias towards building damage assessment. Moreover, the success of our model on both the KumamotoEQ and HaitiEQ datasets, compared to existing methods, indicated that our model could learn to identify damage across a wide variety of building types and materials. Consequently, the strong performance of SurfaceNet demonstrated the value of SurfaceConv and its unique inductive bias.

Additionally, the failure of the pre-trained model indicated that transfer-learning from general point cloud datasets may not be a cure-

all on niche datasets, with the pretrained model not substantially outperforming its unsupervised version. As previously discussed, this is not surprising; large datasets for pre-training of point cloud models are often similar to ShapeNet, a dataset synthetically generated from 3D models which does not have the unique characteristics of aerial LIDAR detailed in Section 2.2.1. The failure of otherwise strong pre-trained models for building damage assessment refutes the argument that pre-trained foundation models, which dominate natural language processing and computer vision, invalidate the careful design of models. Until a significant urban aerial lidar dataset is developed to pre-train models, we speculate that it is unlikely that transfer-learning from point clouds of synthetic 3D models or indoor scenes will be effective when testing aerial LIDAR data.

Overall, for post-earthquake recovery efforts, the success of SurfaceNet in improving the performance of building damage assessors can aid the search and rescue teams and help save lives. For the study of point cloud networks, the success of SurfaceNet suggests that the application of general point cloud networks designed for multiview object datasets to aerial LIDAR point clouds may not be ideal, and that dedicated networks for aerial LIDAR may be necessary.

**4.2** Limitations and perspectives

One substantial limitation of this work was the make-up of the datasets. Due to data access challenges, we could only perform training and testing on the KumamotoEQ and HaitiEQ datasets; consequently, the model may not perform similarly on other datasets. Additionally, the difference in the segmentation labeling between the HaitiEQ and the KumamotoEQ datasets meant that we could not examine the performance of a trained SurfaceNet on out-of-distribution data, for example, the performance of a SurfaceNet model trained on KumamotoEQ and evaluated on HaitiEQ. This segmentation labeling difference arose from two main factors. First, the building damage labeling schemes were different – while labels that the Geospatial Institute of Japan contained 7 classes, the labels on the HaitiBRD dataset only contained four. We could not identify a method to directly translate between these two schemes. Second, the two datasets disagreed on what should be identified as a "building" point – while KumamotoEQ's labels were derived from building footprint maps and consequently considered minor streets as "background" points, HaitiEQ's labels, derived from HaitiBRD, considered the streets between buildings as "building" points. Consequently, the performance transfer of LIDAR building damage detectors across datasets is an area which future studies could further explore.

The overall quality of the segmentation labeling limited this work as well. Labeling point clouds for segmentation involves assigning a class (in this case, background, damaged, or undamaged) to each individual point. In this paper, for the KumamotoEQ dataset, we used historical building polygons from the Geospatial Institute of Japan (18) to algorithmically label the points. However, slight inaccuracies in the specific geolocations of building polygons could result in corresponding inaccuracies in the dataset labeling; for instance, the edges of roofs could be labeled as being background points, or the

ground surrounding a building could be labeled as damaged/undamaged building points. A similar challenge arose in the labeling of the HaitiEQ dataset, albeit to a lesser extent. For the KumamotoEQ dataset, the inaccuracies in dataset labeling explained the substantial difference in performance between the classification models and the patch segmentation models, even though these were tasks that were similar. Future improvements in the segmentation models may require relabeling the point clouds or acquiring more accurate building polygons to improve data quality.

Another limitation of this paper was the challenge of fair comparison across different models' hyperparameter schemes. Different models can have different optimal hyperparameters. For instance, in terms of optimizers, PointMLP natively uses SGD with momentum (27), PointNet$^{++}$ uses Adam (26), and PointNeXt uses AdamW (28). Each of the three models additionally have different native learning rates. However, native hyperparameters may not necessarily be optimal, given that none of the previous models have been trained on the Kumamoto Earthquake Dataset previously. Moreover, as PointNeXt demonstrated, older models such as PointNet$^{++}$ are at a disadvantage simply due to the development of new and more advanced data augmentation schemes, among other hyperparameter modernizations (28). Following DS-Net (10), we simply chose to use the same hyperparameter scheme for all experiments, and tuned hyperparameters only on the validation set. Note also that we selected the best performing checkpoint in the validation runs and assessed that under the test set.

Future studies could further explore aerial LIDAR point cloud-specific networks. One approach to this is by designing aerial LIDAR specific architectures, similar to this paper's approach, which could leverage its unique properties. Another method to improve the performance of point cloud networks on aerial LIDAR data could be the application of pre-existing self-supervised learning methods for point cloud networks to large scale unlabeled aerial LIDAR datasets, such as ReCon (32) or ULIP (31). Self-supervised learning has shown success for computer vision (33), natural language (34), and indeed, for point clouds (31, 32, 35). In this particular case, the fine tuning dataset (the Kumamoto Earthquake Dataset), and the dataset that the pretrained model used for self-supervised learning, were substantially different; however, large scale pretraining on aerial LIDAR datasets may yield substantial improvements.

Finally, this paper only explored one aspect of the search and rescue process; we do not propose a comprehensive search and rescue methodology but a technological tool that can help improve part of the process. As stated in section 1, "Introduction", building damage assessment can aid in parts of two of five steps of the INSARAG rescue process; however, given that building damage assessment or predictions, such as the PAGER software, are already part of the post-earthquake rescue process (36), more effective and efficient methods of building damage assessment can be simply integrated into the existing workflow. Consequently, rescuers may rely upon both damage assessments produced by SurfaceNet as well as other existing data present in PAGER (such as ground movement and

hospital/highway locations) and other similar workflows.

## 5. Conclusion

Overall, effective rapid building damage assessment can be highly valuable for post-disaster search and rescue, with aerial LIDAR being an easy to collect and high detail data format for doing so. We have proposed a novel architecture, SurfaceNet for detecting building damage from aerial LIDAR, based on our SurfaceConv local aggregator. SurfaceConv emphasized modeling intra-surface relationships as an inductive bias for building damage assessment from aerial LIDAR. SurfaceNet achieved state-of-the-art results, outperforming generalized supervised DNNs, pre-trained DNNs, and building-damage specific DNNs.

## 6. Code repository

The code for this paper is available at https://github.com/cory-fan/surfacenet.git

## 7. Abbreviations

LIDAR: Light Detection and Ranging, DNN: Deep Neural Network, CNN: Convolutional Neural Networks, MLP: Multi-Layer Perceptron, KNN: K-Nearest-Neighbor, IOU: Intersection-Over-Union, mIOU: Mean Intersection-Over-Union, SGD: Stochastic Gradient Descent

## 8. References

1. Plank S. Rapid Damage Assessment by Means of Multi-Temporal SAR — a Comprehensive Review and Outlook to Sentinel-1. Remote Sensing, 6: 4870-4906, 2014. https://doi.org/10.3390/rs6064870

2. INSARAG. INSARAG Guidelines 2020. 2020. https://www.insarag.org/methodology/insarag-guidelines/

3. Moltchanova E. Khabarov N. Obersteiner M. Ehrlich D. Moula M. The value of rapid damage assessment for efficient earthquake response. Safety Science, 49: 1164-1171, 2011. https://doi.org/10.1016/j.ssci.2011.03.008

4. Hakami A. Kumar A. Shim SJ. Nahleh YA. Application of Soft Systems Methodology in Solving Disaster Emergency Logistics Problems. International Journal of Industrial Science and Engineering, 7(12): 783-790, 2013.

5. Porter KA. Jaiswal KS. Wald D. Greene M. Comartin C. WHE-PAGER Project: A New Initiative in Estimating Global Building Inventory and its Seismic Vulnerability. The 14th World Conference on Earthquake Engineering, 2008. https://www.iitk.ac.in/nicee/wcee/article/14_S23-016.pdf

6. Copernicus EMS.https://emergency.copernicus.eu/mapping/system/files/components/EMSR546_AOI01_GRA_MONIT54_r1_RTP02_v1.pdf

7. Xu JZ. Lu W. Li Z. Khaitan P. Zaytseva V. Building Damage Detection in Satellite Imagery Using Convolutional Neural Networks. ArXiv, 2019. https://doi.org/10.48550/arXiv.1910.06444

8. Kaur N. Lee C. Mostafavi A. Mahdavi-Amiri A. Large-scale Building Damage Assessment Using a Novel Hierarchical Transformer Architecture on Satellite Images. Computer-Aided Civil and Infrastructure Engineering, 38: 2072-2091, 2023. https://doi.org/10.1111/mice.12981

9. Liu C. Sui H. Huang L. Identification of Building Damage from UAV-Based Photogrammetric Point Clouds Using Supervoxel Segmentation and Latent Dirichlet Allocation Model. Sensors, 20(22): 6499, 2020. https://doi.org/10.3390/s20226499

10. Xiu H. Liu X. Wang W. Kim KS. Shinohara T. Chang Q. Matsuoka M. DS-Net: A Dedicated Approach for Collapsed Building Detection from Post-Event Airborne Point Clouds. International Journal of Applied Earth Observation and Geoinformation, 116, 2023. https://doi.org/10.1016/j.jag.2022.103150

11. Xiu H. Shinohara T. Matsuoka M. Inoguchi M. Collapsed Building Detection Using 3D Point Clouds and Deep Learning. Remote Sensing, 12(24): 4057, 2020. https://doi.org/10.3390/rs12244057

12. Zahs V. Anders K. Kohns J. Stark A. Höfle B. Classification of structural building damage grades from multi-temporal photogrammetric point clouds using a machine learning model trained on virtual laser scanning data. International Journal of Applied Earth Observation and Geoinformation, 122, 2023. https://doi.org/10.1016/j.jag.2023.103406

13. Yamazaki F. Suto T. Matsuoka M. Horie K. Inoguchi M. Liu W. Statistical Analysis of Building Damage in Japan based on the 2016 Kumamoto Earthquake. 17th U.S.-Japan-New Zealand Workshop on the Improvement of Structural Engineering and Resilience Queenstown, New Zealand, 2018. https://pdfs.semanticscholar.org/78cd/cc5c85779ce1a54a05cc46f6203111a1b5a5.pdf

14. Geospatial Institute of Japan. Polygonal Map of Kengun Area (Zone 493016). 2016. https://www.gsi.go.jp/kankyochiri/gm_japan_e.html

15. Rathje EM. Bachhuber J. Dulberg R. Cox BR. Kottke A. Wood CM. Green RA. Olson S. Wells D. Rix G. Damage Patterns in Port-au-Prince during the 2010 Haiti Earthquake. Earthquake Spectra, 27:1-21, 2011. https://doi.org/10.1193/1.3637056

16. OpenTopography. Post-Kumamoto Earthquake (16 April 2016) Rupture Lidar Scan. 2018. https://doi.org/10.5069/G9SX6B9T

17. Scott CP. Arrowsmith JR. Nissen E. Lajoie L. Maruyama T. Chiba T. The M7 2016 Kumamoto, Japan, Earthquake: 3-D Deformation Along the Fault and Within the Damage Zone Constrained From Differential Lidar Topography. Journal of Geophysical Research: Solid Earth, 123(7): 6138-6155, 2018. https://doi.org/10.1029/2018JB015581

18. Moya L. Mas E. Koshimura S. Yamakazi F. Synthetic building damage scenarios using empirical fragility functions: A case study of the 2016 Kumamoto earthquake. International Journal of Disaster Risk Reduction, 31: 76-84, 2018. https://doi.org/10.1016/j.ijdrr.2018.04.016

19. Okada S. Takai N. Classification of Structural Types and Damage Patterns of Buildings for Earthquake Field Investigation. Journal of Structural and Construction Engineering: 524, 1999. http://dx.doi.org/10.3130/aijs.64.65_5

20. Lin S. Edocia M. Tsai FJ. Chen L. Kuo WN. HaitiBRD: A labeled satellite imagery dataset for building and road damage assessment of the 2010 Haiti earthquake. DesignSafe-Cl. 2023. https://doi.org/10.17603/ds2-fqat-4v02

21. OpenTopography. World Bank – ImageCat, Inc. - RIT Haiti Earthquake Lidar dataset. 2021. https://doi.org/10.5069/G96Q1V50

22. DesRoches R. Comerio M. Eberhard M. Mooney W. Rix GJ. Overview of the 2010 Haiti Earthquake. Earthquake Spectra, 27: S1-S21, 2011. https://escweb.wr.usgs.gov/share/mooney/142.pdf

23. Manzini T. Perali P. Karnik R. Murphy R. CRASAR-U-DROIDs: A Large Scale Benchmark Dataset for Building Alignment and Damage Assessment in Georectified sUAS Imagery. Arxiv Preprint, 2024. http://dx.doi.org/10.48550/arXiv.2407.17673

24. Nong X. Bai W. Liu G. Airborne LiDAR Point Cloud Classification using PointNet++ Network with Full Neighborhood Features. PLoS ONE, 18(2), e0280346, 2023. https://doi.org/10.1371/journal.pone.0280346

25. Qi CR. Yi L. Su H. Guibas LJ. PointNet$^{++}$: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. 31st Conference on Neural Information Processing Systems, 2017. https://proceedings.neurips.cc/paper_files/paper/2017/file/d8bf84be3800d12f74d8b05e9b89836f-Paper.pdf

26. Axel C. van Aardt J. Building Damage Assessment using Airborne Lidar. Journal of Applied Remote Sensing, 11, 2017. http://dx.doi.org/10.1117/1.JRS.11.046024

27. Ma X. Qin C. You H. Ran H. Fu Y. Rethinking Network Design and Local Geometry in Point Cloud: A Simple Residual MLP Framework. International Conference on Learning Representations, 2022. https://doi.org/10.48550/arXiv.2202.07123

28. Qian G. Li Y. Peng H. Mai J. Hammoud H. Elhoseiny M. Ghanem B. PointNeXt: Revisiting PointNet$^{++}$ with Improved Training and Scaling Strategies. 36th Conference on Neural Information Processing Systems, 2022. https://doi.org/10.48550/arXiv.2206.04670

29. Qi C. Su H. Mo K. Guibas LJ. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. IEEE Conference on Computer Vision and Pattern Recognition, 2017. https://doi.org/10.1109/CVPR.2017.16

30. Wang Y. Sun Y. Liu Z. Sarma S. Bronstein M. Solomon J. Dynamic Graph CNN for Learning on Point Clouds. ACM Transactions on Graphics, 38(5):1-12, 2019. https://doi.org/10.1145/3326362

31. Xue L. Gao M. Xing C. Martin-Martin R. Wu J. Xiong C. Xu R. Niebles JC. Savares S. ULIP: Learning a Unified Representation of Language, Images, and Point Clouds for 3D Understanding. IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023. https://openaccess.thecvf.com/content/CVPR2023/papers/Gao_ULIP_Learning_a_Unified_Representation_of_Language_Images_and_Point_CVPR_2023_paper.pdf

32. Qi Z. Dong R. Fan G. Ge Z. Zhang X. Ma K. Yi L. ReCon: Contrast with Reconstruct: Contrastive 3D Representation Learning Guided by Generative Pretraining. International Conference on Machine Learning, 2023. https://proceedings.mlr.press/v202/qi23a/qi23a.pdf

33. He K. Chen X. Xie S. Li Y. Dollár P. Girshick R. Masked Autoencoders Are Scalable Vision Learners. IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022. https://openaccess.thecvf.com/content/CVPR2022/html/He_Masked_Autoencoders_Are_Scalable_Vision_Learners_CVPR_2022_paper.html

34. Devlin J. Chang MW. Lee K. Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. North American Chapter of the Association for Computational Linguistics: 4171-4186, 2019. https://doi.org/10.18653/v1/N19-1423

35. Yu X. Tang L. Rao Y. Huang T. Zhou J. Lu J. Point-BERT: Pre-training 3D Point Cloud Transformers with Masked Point Modeling. IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022. https://openaccess.thecvf.com/content/CVPR2022/papers/Yu_Point-

BERT_PreTraining_3D_Point_Cloud_Transformers_With_Masked_Point_Modeling_CVPR_2022_paper.pdf

36. Wald D. ShakeMap & ShakeCast for Earthquake Planning & Response. Earthquake Summit 2022, 2022. https://sema.dps.mo.gov/earthquake_preparedness/docs/Wald.pdf