

Article

# Deep Learning Classification of 2D Orthomosaic Images and 3D Point Clouds for Post-Event Structural Damage Assessment

Yijun Liao, Mohammad Ebrahim Mohammadi  and Richard L. Wood \* 

Department of Civil and Environmental Engineering, University of Nebraska-Lincoln, Lincoln, NE 68588-531, USA; yijun.liao419@huskers.unl.edu (Y.L.); me.m@huskers.unl.edu (M.E.M.)

\* Correspondence: rwood@unl.edu

Received: 15 May 2020; Accepted: 20 June 2020; Published: 22 June 2020



**Abstract:** Efficient and rapid data collection techniques are necessary to obtain transitory information in the aftermath of natural hazards, which is not only useful for post-event management and planning, but also for post-event structural damage assessment. Aerial imaging from unpiloted (gender-neutral, but also known as unmanned) aerial systems (UASs) or drones permits highly detailed site characterization, in particular in the aftermath of extreme events with minimal ground support, to document current conditions of the region of interest. However, aerial imaging results in a massive amount of data in the form of two-dimensional (2D) orthomosaic images and three-dimensional (3D) point clouds. Both types of datasets require effective and efficient data processing workflows to identify various damage states of structures. This manuscript aims to introduce two deep learning models based on both 2D and 3D convolutional neural networks to process the orthomosaic images and point clouds, for post windstorm classification. In detail, 2D convolutional neural networks (2D CNN) are developed based on transfer learning from two well-known networks AlexNet and VGGNet. In contrast, a 3D fully convolutional network (3DFCN) with skip connections was developed and trained based on the available point cloud data. Within this study, the datasets were created based on data from the aftermath of Hurricanes Harvey (Texas) and Maria (Puerto Rico). The developed 2DCNN and 3DFCN models were compared quantitatively based on the performance measures, and it was observed that the 3DFCN was more robust in detecting the various classes. This demonstrates the value and importance of 3D datasets, particularly the depth information, to distinguish between instances that represent different damage states in structures.

**Keywords:** convolutional neural network; deep learning; transfer learning; point clouds; structural damage assessment

---

## 1. Introduction

One of the emerging approaches for aerial image collection is to utilize the unpiloted (or unmanned) aerial system (UAS), commonly known as a drone [1–3]. Following natural hazard events, the data collection is often limited by time and site accessibility imposed by precarious structures, debris, road closures, curfews, and other restrictions. However, UAS imagery enables first responders and emergency managers to perform effective logistical planning, loss estimates, and infrastructure assessment for insurance adjusters, engineers, and researchers [4]. UAS with an onboard camera enables assessors to collect numerous images from large areas efficiently as well as to reconstruct the three-dimensional (3D) scene via three steps, including Scale Invariant Feature Transform (SIFT), Structure-from-Motion (SfM), and Multi-View Stereo (MVS). Here, SfM is generated using two-dimensional (2D) aerial images [5]. The SfM derived point cloud has relative accuracy at the

centimeter level [1]. The creation of a 3D SfM point cloud is a time-consuming process; however, it enables a reconstruction of the depth information in the scene, which may improve various analyses. Recently, deep learning techniques have become a more common approach to develop various computer vision workflows. These techniques have been utilized to create workflows to investigate damage in the aftermath of extreme events from aerial images, in particular for large areas and at the community scale.

The main objective of this manuscript is to study and compare three deep learning models based on 2D aerial images and 3D SfM derived point clouds to detect damaged structures following two hurricanes. In addition, the study investigates the application of transfer learning for the 2D convolutional neural network (CNN) as a rapid post-event strategy to develop a model for damage assessment of built-up areas with minimal to no prior data. The model created for damage assessment using 2D images is developed based on transfer learning from two well-known image classification networks, namely AlexNet and VGGNet [6]. During the training process, the pre-trained weights of these models were further modified to match the user-defined classes. Moreover, a 3D fully convolutional network (3D FCN) with skip connections is developed based on expanding the 3D FCN model proposed by Mohammadi et al. [7]. While the goal of 2DCNN is to classify the aerial images based on the most prominent object observed in the images, the 3D FCN with skip connections semantically classifies the SfM derived point cloud. Both 2D and 3D models were trained on similar datasets with identical numbers of classes and were compared based on precision, recall, and overall accuracy. The comparison between 2D and 3D models is to demonstrate the entropy and value associated with the depth-information that is present in the 3D datasets.

## 2. Literature Review

### 2.1. Studies Used 2D Images for Detection and Classification

The task of object detection or classification of a set of images has been investigated by various studies using CNNs with different architectures. Among all proposed methods, transfer learning has become one of the most popular techniques. Transfer learning corresponds to the process of fine-tuning the upper layers of a pre-trained model based on a new dataset for a newly proposed task(s) [8]. Models developed based on a transfer learning strategy demonstrated not only improved performance in comparison to other models but also that such models could be developed more efficiently. In an early study within the area of deep learning, Bengio discussed transfer learning algorithms and their effectiveness in classifying new instances based on pre-trained models and demonstrating the process through numerous examples of transfer learning [8]. One of the most referenced studies was performed by Krizhevsky et al. [9]. The authors within this study developed a CNN model through transfer learning from the subset of the ImageNet network [10]. Krizhevsky et al. modified the fully connected layers to accommodate classifying the new labels [9]. The modified network architecture contained eight layers to maximize the correct label under prediction distribution. The authors reported that the validation of the results showed test error rates of 17% to 37.5% for the used datasets. Another application of transfer learning was studied by Oquab et al. [11]. The authors developed various CNN models for the task of visual recognition based on transfer learning from the pertained ImageNet model [10]. Within this study, the training images are mainly comprised of centered objects with a clear background in the image, and authors reported that the model was able to classify images with a high level of accuracy after an extended training process. Different CNN performs distinctively as the network architecture varies. As a result, Shin et al. listed a few popular image-based classification transfer learning networks, such as CifarNet, AlexNet, and GoogleNet [6]. These network performances were compared based on a set of medical images via transfer learning. It was concluded that transfer learning has been consistently beneficial for classification experiments.

Various studies have investigated the application of CNN models for post-event assessments using aerial images. For example, Hoskere et al. proposed post-earthquake inspections based on UAS imagery and CNN models [12]. Within this study, the authors developed a fully convolutional

network to semantically segment images into three classes of pixels. The developed model was able to segment the images with an average accuracy of 91.1%. More recently, Xu et al. studied the post-earthquake scene classification task using three deep learning methods. These methods included a Single Shot MultiBox Detector (SSD), post-earthquake multiple scene recognition (PEMSR) based on transfer learning from SSD, and Histogram of Oriented Gradient along with Support Vector Machine (HOG+SVM) [13]. Within the proposed method, the aerial images were initially classified into six classes, including landslide, houses, ruins, trees, clogged, and ponding. The dataset was created via web-searched images of the 2014  $M_W$  6.5 Ludian earthquake (China), which were later preprocessed and degraded into  $300 \times 300$  pixels and manually classified into the six classes aforementioned. The authors reported that the PEMS model demonstrated a higher efficiency of 0.4565 s comparing to HOG+SVM of 8.3472 s as well as higher accuracy. In their work, the transfer learning strategy also improved the overall accuracy and performance, although the average processing time was slightly higher than the SSD method. Moreover, in addition to the effect of transfer learning in the increasing accuracy and performance of 2D CNN models, Simonyan and Zisserman pointed out the CNN performance improvement can be achieved by increasing the network depth [14]. As a result, Gao and Masalam developed a deep 2D CNN based on transfer learning from the VGGNet model for Structural Health Monitoring (SHM) and rapid post-event damage detection [15]. The 2D image-based SHM used red, green, blue (RGB) information based unsupervised training algorithms and was able to obtain 90% accuracy for binary classification.

## 2.2. Studies Used 3D Point Clouds for Detection and Classification

With the rapid development of technologies to collect remotely sensed 3D point clouds and the growing application of these data in various fields of civil engineering, many researchers have proposed various methods to analyze 3D point clouds, in particular for routine inspections or post-event data collection and analyses [16,17]. The datasets here are considered to be non-temporal, which is a single post-event only dataset that does not utilize change detection from a baseline (or pre-event) dataset. For example, Axia et al. proposed a workflow to classify an aerial 3D point cloud into damaged and undamaged classes [18]. Within the proposed workflow, Axia et al. estimated a normal vector for each point within the point cloud data as the key damage sensitive feature and identified the variation of these normal vectors with respect to a global reference vector. Lastly, the study used a region growing approach based on the variation of normal vectors to classify the point cloud. Axia et al. reported that while the proposed method can classify the collapsed structures, the developed method may misclassify the partially damaged structures.

In general, one of the main steps in point cloud analysis workflows is to classify the points into a set of predefined classes. As a result, multiple workflows have been introduced to classify point clouds through machine learning and more recently deep learning techniques. Hackel et al. introduced one of the most successful workflows to classify dense point clouds of urban areas into multiple classes [19]. These classes include building façades, ground, cars, motorcycles, traffic signals, and pedestrians. Within this study, authors extract a series of features for each point based on various neighboring sizes using eigendecomposition, the height of points, and first and second statistical moments and used random forest learning algorithm to classify each point. The proposed method results in the main overall accuracy of 95%. More recently, Xing et al. used the Hackel et al. workflow as a basis and developed a more robust workflow by adding a series of features computed based on the difference of normal vectors for better identification [19,20]. Their study demonstrated a 2% improvement on average.

Recently deep learning method applications become more widespread to analyze 3D datasets in science and engineering fields. Various deep learning-based workflows have been developed to classify the 3D point cloud datasets. The main advantage of deep learning-based learning algorithms over the more traditional learning algorithms (e.g., artificial neural networks) is the capability of the algorithms to learn the feature extractors from the input data directly. Therefore, deep learning algorithms,

in particular CNN architectures, eliminate the need for engineering feature extractors based on the geometry of the objects within the dataset and background. One of the early studies to investigate the application of deep learning for 3D point cloud classification was performed by Prokhorov [21]. Prokhorov proposed a 3D network architecture similar to CNN to classify point cloud of various objects by converting the point cloud data into 3D grid representations. The developed 3D CNN network had one convolutional layer, one pooling layer, and two fully connected layers, which was followed by a 2-class output layer. The weights or parameters within the convolutional layers were pre-trained using lobe component analysis and were updated using the stochastic meta-descent method [22]. Following this study, Maturana and Scherer proposed a 3D CNN for object recognition similar to that of Prokhorov [22,23]. The proposed 3D network had two tandem convolutional layers, one max-pooling layer, and one fully connected layer, which was followed by the output layer. In contrast to the study conducted by Prokhorov [22], Maturana and Scherer did not pre-train the developed network while the network resulted in a performance on par or better than the network proposed by Prokhorov. This highlights that the developed 3D CNN network was able to extract features during the training process effectively.

Recently, Hackel et al. introduced a point cloud classification network based on 3D CNN architecture. The proposed network accepts five occupancy grid models with different resolutions for each instant as input and has five convolutional layers in parallel with an organization similar to VGGNet, which followed by a series of fully connected and one output layers [14,24]. The authors have reported a maximum overall accuracy of 88% and an intersection over the union value of 62% for datasets collected from urban environments. This work classifies the scene into classes of natural terrain, high vegetation, low vegetation, buildings, hardscape, vehicles, and human-made terrains. More recently, Zhang et al. proposed a network to semantically segment point clouds based on a model that consists of three distinct networks [25]. The first network encodes the point cloud into 2D instances. The second network consists of a series of fully connected and max-pooling layers, which are followed by convolutional layers. Finally, the third and last network goal converts the 2D encoded data into 3D grid models, which semantically classify the voxels in the grid and creates a bounding box for each detection object. The authors reported that the experimental results demonstrate an overall improvement in accuracy of 10% in comparison to the network developed by Maturana and Scherer [25].

### 2.3. Knowledge Gap

Previous studies have explored the application of CNNs in post-natural hazard event assessment using aerial images. Both deep learning-based methods and unsupervised learning were implemented for 2D and 3D datasets, while the difference between 2D and 3D datasets in deep learning has yet to be fully understood by quantitative comparisons. As reviewed, the majority of studies developed to analyze the 3D point clouds for post-event applications were created based on traditional methods. In contrast, the applications of deep learning models developed based on transfer learning for 2D aerial images was investigated in various studies. However, due to the lack of depth information, limitations of the damage and structural component recognition still exist. As a result, this study investigates the application of deep learning-based models using 2D images and 3D SfM derived point clouds corresponding to the same post-event scenes.

## 3. Datasets

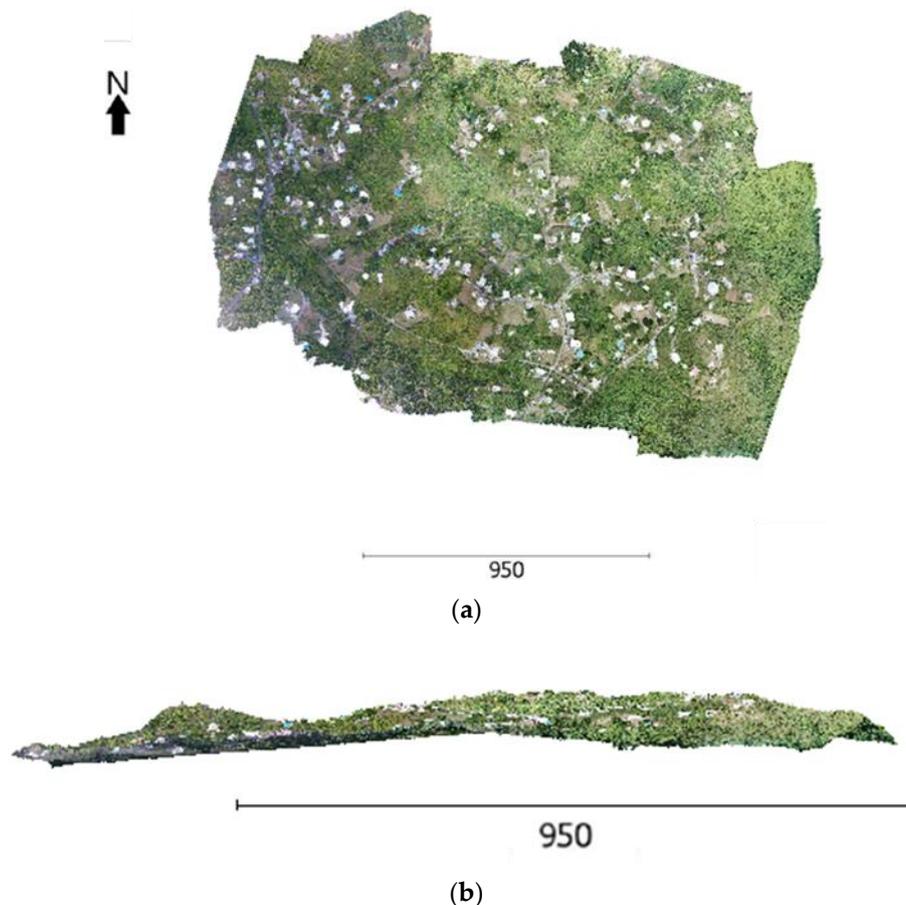
### 3.1. Introduction to Hurricane Harvey and Maria

Within this study, three othomosiac image and point cloud datasets were collected in the aftermath of Hurricanes Harvey and Maria. Hurricane Harvey made landfall on 25 August 2017 on the coastline of Texas. Hurricane Harvey was a Category 4 hurricane and produced wind gusts over 215 km/h, and storm surges as high as 3.6 m. This incident resulted in the destruction of more than 15,000 partial

damage, 25,000 residential and industrial structures, as well as other critical infrastructure in coastal communities, including the towns of Rockport and Port Aransas [26]. Hurricane Maria made landfall on 19 September of 2017 in Puerto Rico. Hurricane Maria was classified as Category 5 hurricane and produced wind gusts over 280 km/h, and storm surges as high as 2.3 m, which makes it the most severe natural hazard event recorded in history to affect Puerto Rico and other Islands in the region [27]. As a result of this extreme event, the power grid of Puerto Rico was significantly damaged, a major dam for the Guajataca reservoir sustained critical structural damage, and more than 60,000 buildings were damaged [28].

### 3.2. Data Collection Method

To carry out the data collection task of the selected areas, a medium-size drone with an onboard camera was deployed. A DJI Phantom 4 UAS collected high-resolution aerial images with an onboard camera. The selected flight paths were fully controlled autonomously with the Pix4dcapture application on a handheld tablet. The data collection in Puerto Rico produced 4077 images in 7 flights, which covered approximately 1.75 km<sup>2</sup> area with a 53.5 m elevation change. The Texas Salt Lake dataset contained 1379 images in 2 flights with 0.75 km<sup>2</sup> area coverage with an elevation range of 9.3 m. The Texas Port Aransas site had 1424 images collected from four flights, with a 0.88 km<sup>2</sup> area coverage with an elevation range of 1.9 m. The collected images were further processed using SfM workflow, which used a series of two-dimensional images with sufficient overlap to generate 3D point cloud and further processed orthomosaic datasets of the surveyed area [1]. The SfM derived point clouds for the two sites are shown in Figures 1–3. Other key characteristics of these datasets are presented in Table 1.



**Figure 1.** Puerto Rico point cloud (scale in meters): (a) top view and (b) side view.

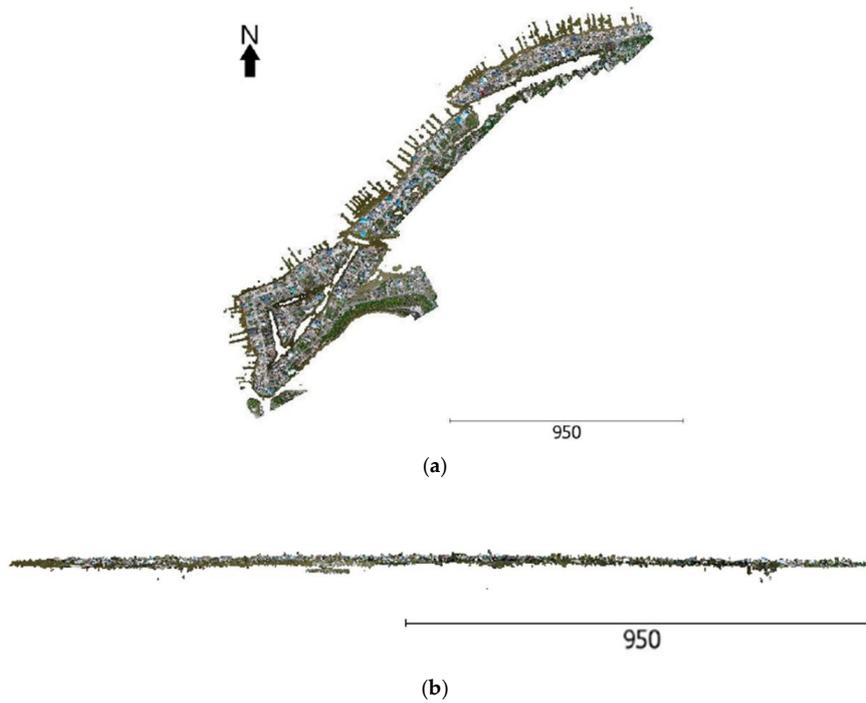


Figure 2. Texas Salt Lake point cloud (scale in meters): (a) top view and (b) side view.

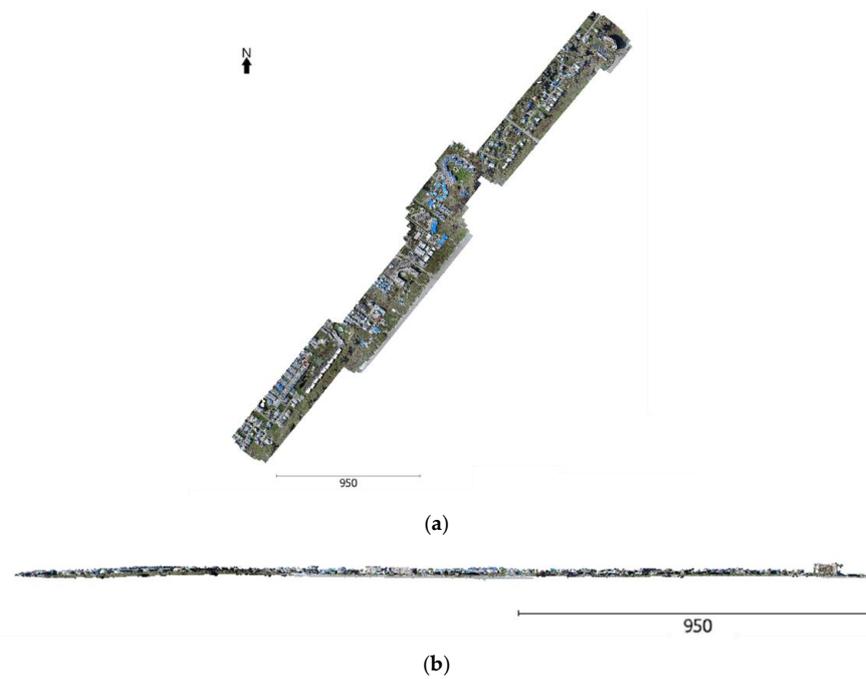


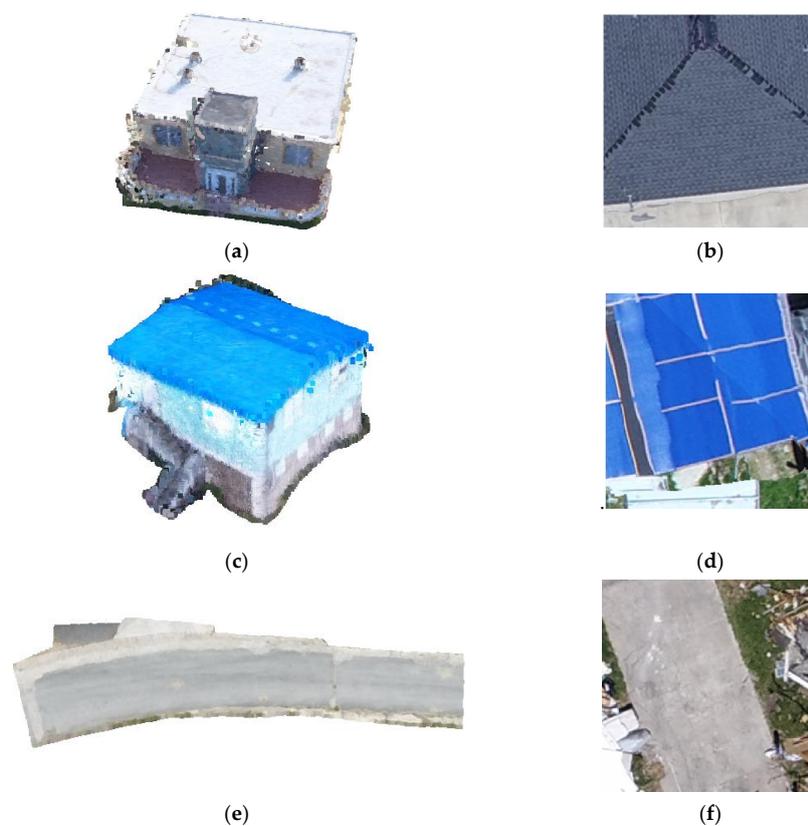
Figure 3. Texas Port Aransas point cloud (scale in meters): (a) top view and (b) side view.

Table 1. Summary characteristics of the datasets.

Dataset	Characteristics		
	GSD (cm)	Orthomosaic Dimensions (pixels)	Point Cloud Number of Vertices (count)
Puerto Rico	1.09	29,332 × 39,482	393,764,295
Texas—Salt Lake	2.73	61,395 × 61,937	78,830,950
Texas—Port Aransas	2.69	96,216 × 84,611	131,902,480

### 3.3. Dataset Classes

Within this study, each dataset was segmented manually into one of the following seven classes: Undamaged structures, partially damaged structures, completely damaged structures, debris, roadways, terrain, and vehicles. An earlier study by Mohammadi et al. informs the classification used here [7]. However, the scope of damaged structures is expanded where the instances are modified into two damaged structure classes based on the level of damage sustained during the event, which relates to the degree of damage sustained. A structure that sustained partial damage includes any building that does not represent any physical changes while the roof of the structure is covered by tarps, which are typically blue or red. Completely damaged structures are buildings that underwent physical changes due to the event, such as roof damage without tarp coverings with visible structural components such as beams, columns, or walls. If a structure is collapsed such that no structural component can be identified, the structure is classified as debris. The class of debris consists of everything that is not in its native state. Debris, in general, is comprised of rooftop shingles, fallen trees, downed utility or light poles, and other wind-blown objects. Terrain incorporates any region that is comprised of grass, low-height vegetation, water, sand, trees, exposed soil, fences, or utility poles. Note that any nonbuilding structural objects that are represented by a cylindrical shape (e.g., utility and light poles) are considered as terrain [29]. Lastly, the vehicle class corresponds to objects used for the transportation of people or goods. This includes cars, SUVs, trucks, carts, recreational vehicles, trailers, construction vehicles (e.g., excavators), or any water-borne vessels that can be propelled on water by oar, sail, or engine. Figures 4 and 5 demonstrate the example of each class point cloud and its corresponding images. Lastly, Tables 2 and 3 illustrate the number of point cloud and image instances that were manually identified from the datasets, respectively.



**Figure 4.** Examples of instances from all datasets: (a,b) Undamaged structure, (c,d) partially damaged structure, and (e,f) roadways.



**Figure 5.** Examples of instances existed from all datasets: (a,b) vehicles, (c,d) terrain, (e,f) debris field, and (g,h) completely damaged structure.

**Table 2.** Summary of point cloud instances for Salt Lake, Puerto Rico, and Port Aransas.

Instance	Number of Instances		
	Texas-Salt Lake	Puerto Rico	Texas-Port Aransas
Terrain	719	224	665
Undamaged Structure	307	97	355
Debris	404	764	257
Partially Damaged Structure	99	76	115
Completely Damaged Structure	146	364	76
Vehicle	256	198	224
Roadway	57	166	87

**Table 3.** Summary of image instances for Salt Lake, Puerto Rico, and Port Aransas.

Instance	Number of Instances		
	Texas-Salt Lake	Puerto Rico	Texas-Port Aransas
Terrain	1972	5238	3288
Undamaged Structure	138	610	688
Debris	296	247	71
Partially Damaged Structure	236	223	485
Completely Damaged Structure	152	74	33
Vehicle	67	160	53
Roadway	246	864	904

## 4. Methodology

### 4.1. Dataset Preparation for 2D Images

The process of creating image instances was started by creating an orthomosaic image of the entire scene. This was done using Pix4Dmapper. Afterward, the orthomosaic image was segmented into a series of  $256 \times 256$  images. As a result, approximately a total of 18,000 images were created from the Puerto Rico dataset, the Salt Lake dataset resulted in a total of 60,000 images, and Port Aransas dataset was divided into 120,000 segmented images. The next step within the preparation image instances was to assign a label to each  $256 \times 256$  image based on the seven classes mentioned in Section 3. Within this study, the image classes are determined by the most prominent object that is visible in the image. Moreover, the Salt Lake and Puerto Rico datasets were used for model development, and the Port Aransas dataset was used to test and validate the developed models.

### 4.2. 2D Convolutional Neural Network Architecture

Pre-trained CNNs have advantages due to their relative stability during the training process, efficiency, and higher performance over various diverse tasks. Among the various networks available to select for transfer learning, AlexNet and VGGNet were selected as a basis to develop the 2D CNN models in MATLAB 2020a. These two networks were pre-trained by millions of images for more than 1000 classes. These selected networks for transfer learning, on the other hand, represented different architectures. AlexNet model was developed in 2012 and was the first CNN model to perform well on the ImageNet database, and it still performed consistently well on diverse datasets [9,30]. This network contained five layers, including convolutional and max-pooling layers, and two fully connected layers, as illustrated in Figure 6. The developed model based on AlexNet had identical architecture to the AlexNet network; however, within the fully-connected layers, the dropout regularization method was applied to combat the overfitting while training [31]. The input images were also augmented through rotation and reflection processes to reduce the generalization error of the models. The second CNN model was developed based on transfer learning from VGGNet from 2014. The VGGNet model had 16 convolutional and max-pooling layers, which was followed by the fully connected layers, as shown in Figure 7. These small filter sizes (i.e.,  $3 \times 3$  kernels) in VGGNet captured and learned the small details of input instances while larger filter sizes of the network (i.e.,  $5 \times 5$ ) permitted the network to extract features that corresponded to larger regions. Development of the networks based on transfer learning permitted to modify the previously learned feature extractors of these networks according to a new task using a smaller number of training images and epochs [30].

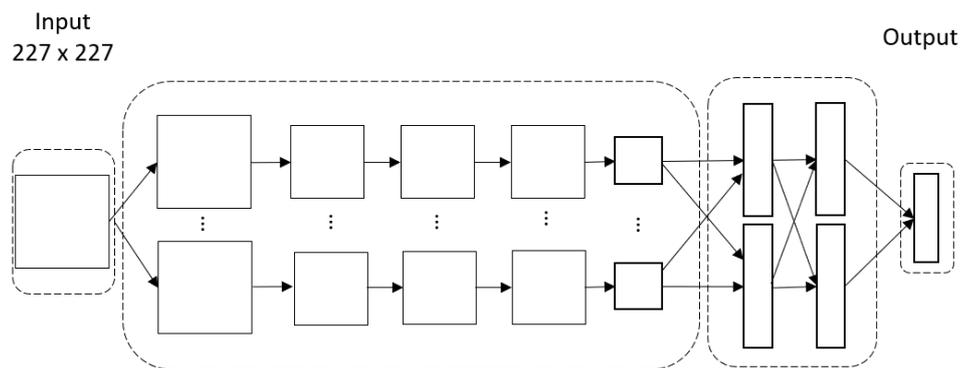


Figure 6. AlexNet network architecture.

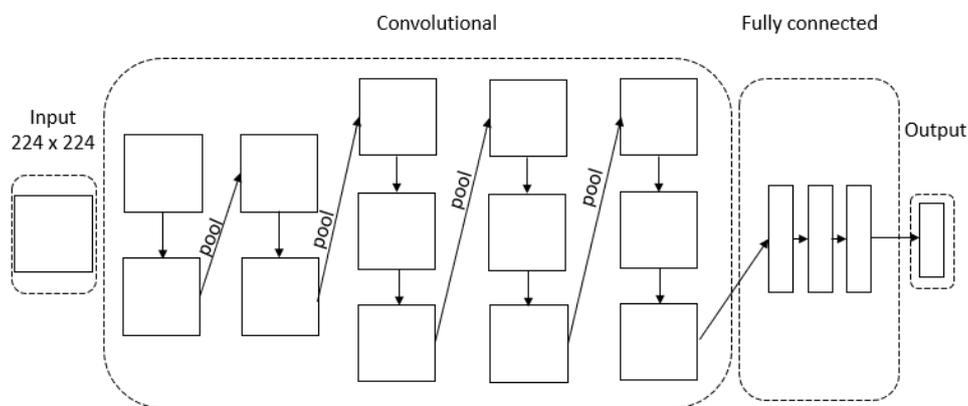
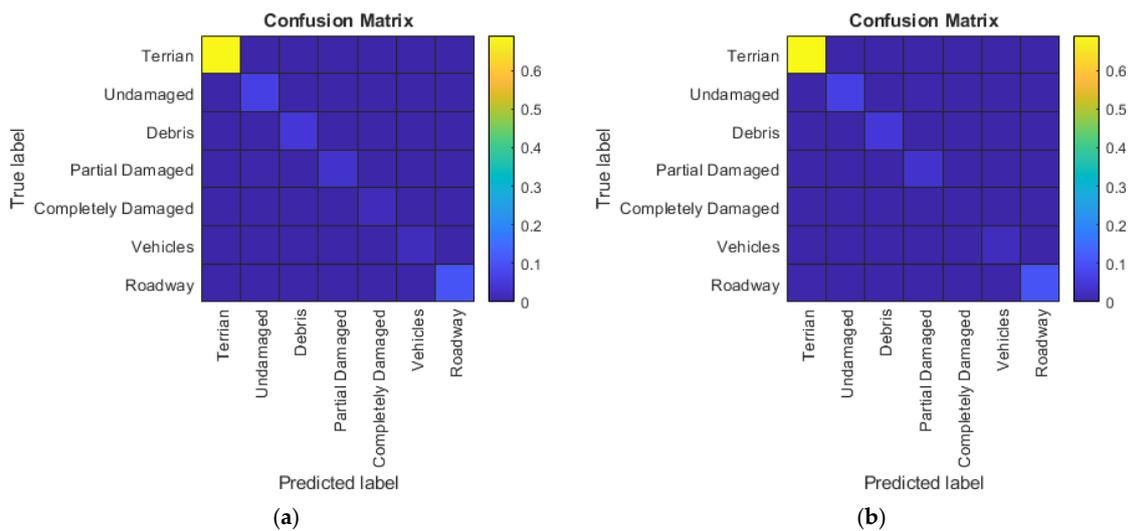


Figure 7. VGGNet network architecture.

During the training process of models developed based on the transfer learning strategy, the  $256 \times 256$  image instances were rescaled to  $227 \times 227$  and  $224 \times 224$  for AlexNet and VGGNet, respectively. In addition, the batch size, which represents the number of images input into the network at once, was set to 64 [15]. While the number of epochs was originally set as high as 2000, the training was terminated when the computed losses reached a plateau to combat the overfitting. The learning rate was set as 0.01 for both networks. Besides these parameters, the remaining hyperparameters were kept identical to the original networks [32]. Note that the training images for both AlexNet and VGGNet are identical in order to compare the results, and because of the augmentation process, approximately over 10,000 images within seven classes are used within the network training. The training performance was evaluated by computing the losses and validation accuracy. Generally, the training of AlexNet contained approximately 300 iterations, while VGGNet has a higher number of approximately 500 iterations. Both developed networks resulted in optimized accuracy for seven classes, which is 88.7% of AlexNet and 91.0% of VGGNet. Figure 8 shows the confusion matrices for the developed networks, which demonstrates that both networks were able to detect the majority of class terrain with high-level accuracy.

The evaluation of the results indicated that the model struggled to learn differences in some of the original classes, particularly as related to structural damage assessment. Consequently, a selected number of classes were merged to reduce the total number of classes from the original seven to specific five and four classes. This was done to demonstrate if the model was able to distinguish between a structural class in general, and an improvement was noted. However, none of the networks were able to learn the other classes, including the partially damaged structure, completely damaged structure, and debris, due to significant similarities between partially damaged and completely damaged structures within the segmented orthoimages.



**Figure 8.** 2D convolutional neural network (CNN) confusion matrices during the training process: (a) AlexNet and (b) VGGNet models.

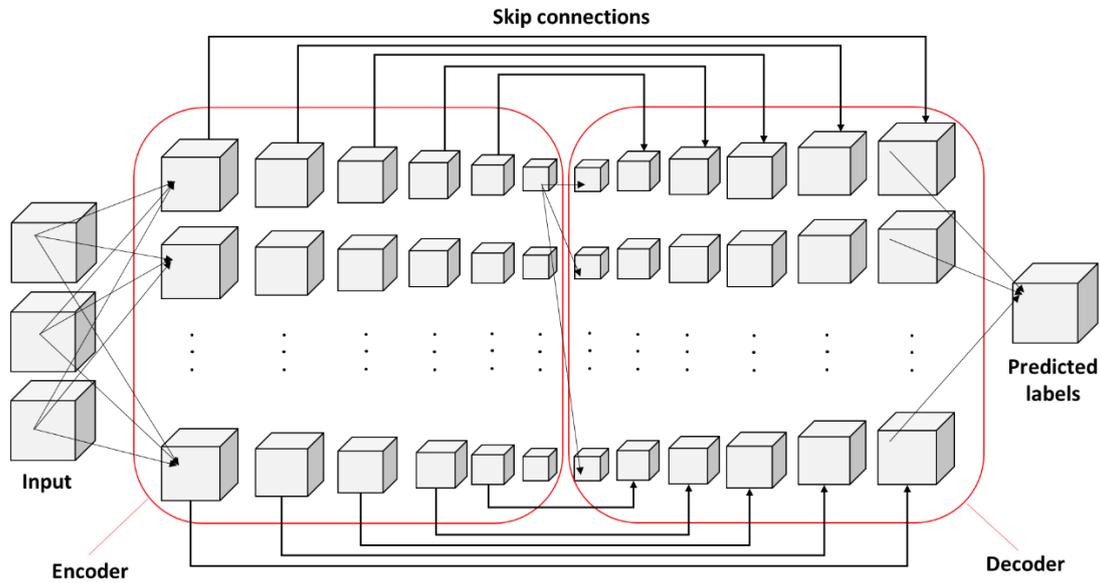
#### 4.3. Dataset Preparation for 3D Point Clouds

Raw and unstructured point clouds were typically incompatible with CNN architectures. This was due to the issue that point clouds generally lack a grid structure, unlike images. Consequently, the raw point cloud instances were converted into volumetric or occupancy grid models, which were 3D arrays. Occupancy grid models provided a suitable data structure for point clouds that can be used within robust CNN learning models. To convert the point clouds instances to occupancy grid models, a method as proposed by Mohammadi et al. was used [7]. Within this study, initially, the point cloud instances were created by slicing the labeled point cloud dataset into roughly  $10\text{ m} \times 10\text{ m}$  segments. Then, the coordinates within each segment, which consisted of objects with various labels, were processed to have only positive values and normalized [7]. Afterward, each segment was downsampled based on the selected occupancy grid dimensions. Within this study, the occupancy grid model of  $64^3$  was used as it results in a sampling of 10 to 16 cm for  $10\text{ m} \times 10\text{ m}$  segments, which was a sufficient resolution to perform per building damage assessment in the aftermath of wind storm events [17]. Lastly, an extra-label corresponding to the empty cells within the 3D arrays was assigned to each instance and denoted as neutral. This allowed the network not only to learn the label instances but also to learn the geometry of the output based on the input instances as well since occlusion or gaps in point clouds are common.

#### 4.4. 3D Fully Convolutional Network Architecture with Skip Connections

The model developed to learn 3D point cloud instances was guided by the previous work of Long et al. and, as discussed in Mohammadi et al. [7,33]. However, the authors reported that developed 3D FCN required a large number of training iterations to achieve an acceptable level of accuracy. As a result, the 3D FCN architecture was modified within this study with skip connections, such that the network can recover the most useful features during the training process at a faster rate [25,34]. The 3D FCN was implemented in TensorFlow v1.15 within this study, and the developed 3D FCN model had an overall general architecture similar to that presented in Mohammadi et al. [7]. In summary, the network was comprised of an input layer that accepted three 3D arrays corresponding to red, green, and blue channels. In addition, the network consisted of an encoding part and decoding parts. The encoder was comprised of 6 3D convolutional layers. The decoding segment of the network consisted of a total of 6 3D transpose convolutional layers. Note that the network did not use any max-pooling layers. Lastly, the output layer was a single-occupancy grid model, each of which cells represented the label of the input point cloud instance (Figure 9). Skip connections added the output

of the convolutional layers within the encoder to the corresponding input of transpose convolutional layers in the decoder. The skip connections conceptually helped the network to recover the fine details in the prediction and reduce any gradient vanishing issues. Figure 9 illustrates the skip connections by arrows.



**Figure 9.** The developed 3D fully convolutional network with skip connections pipeline.

The developed 3D FCN with skip connections was optimized based on stochastic gradient descent, and the cells that contained labels except neutral classes were waited by a factor of 2.0 while updating the learnable parameters to increase boost training and reduce the convergence time. The model was trained on instances from the Salt Lake and Puerto Rico datasets. To further improve the network for generalization, the training instances were augmented by randomly rotating each instance two times. This resulted in a total of 10,958 training instances. In addition, it was observed that network convergence improved as the number of mini-batches increased from 64 to 256. Therefore, the model was trained based on the minibatch size of 256. To evaluate the training process, three performance measures were also calculated, including precision, recall, and cell accuracy in addition to loss, as shown in equations below:

$$Recall = \frac{C_{ii}}{C_{ii} + \sum_{j \neq i} C_{ij}} \quad (1)$$

$$Precision = \frac{C_{ii}}{C_{ii} + \sum_{j=i} C_{ji}} \quad (2)$$

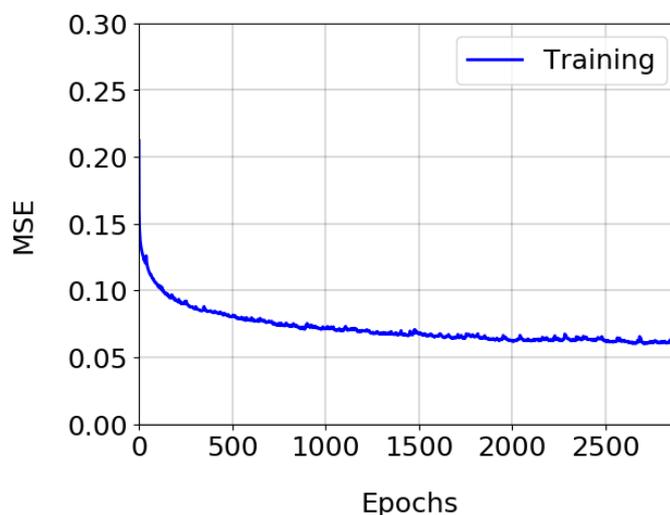
$$Cell\ accuracy = \frac{\sum C_{ii}}{\sum_i \sum_j C_{ji}} \quad (3)$$

where  $C_{ii}$  represents the diagonal of the confusion matrix, which also corresponds to true predictions,  $\sum_{j \neq i} C_{ij}$  denotes to the false negatives,  $\sum_{j \neq i} C_{ji}$  denotes the false positive predictions,  $\sum C_{ii}$  represents the total count of true predictions, and  $\sum_i \sum_j C_{ji}$  represents the total count of all predictions. Table 4 demonstrates the result of these performance measures for the developed model during training for a total of 2500 epochs, and Figure 10 shows the training losses, which was measured based on mean squared error (MSE). Lastly, Figure 11 demonstrates the confusion matrix for the trained model. The training results demonstrated that while the model had learned the geometry of the input instances with a high level of accuracy, cell accuracy of 98.1%, it cannot distinguish the discrepancy between partially damaged structures, completely damaged structures, debris, and vehicles. Lastly,

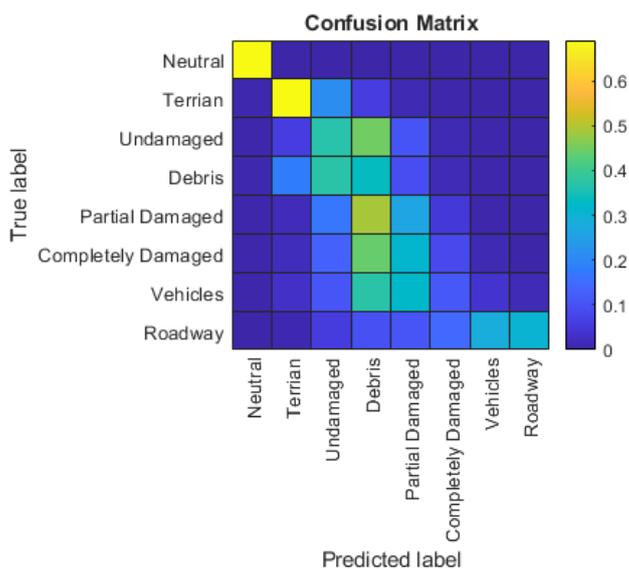
the developed network with skip connections demonstrated massive improvement in comparison to the 3D FCN model introduced by Mohammadi et al. as the network was able to achieve a similar level of accuracy with almost 25% training iterations [7].

**Table 4.** Quantified performance measures for the training dataset.

Classes	3D FCN	
	Precision	Recall
Neutral	100	100
Terrain	85	70
Undamaged Structure	15	37
Debris	31	33
Partially Damaged Structure	8	26
Completely Damaged Structure	14	8
Vehicle	4	4
Roadway	94	94



**Figure 10.** Loss progress (MSE) during model training.

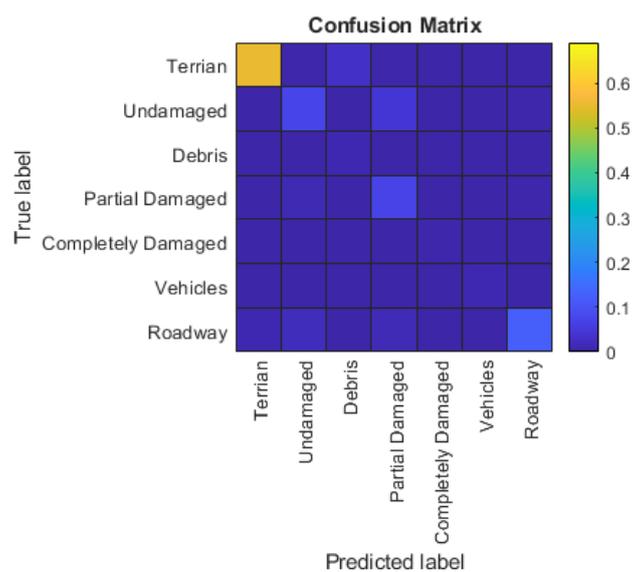


**Figure 11.** 3D fully convolutional network (FCN) confusion matrix for the training dataset.

## 5. Discussion

### 5.1. 2D CNN Experiment

The developed 2D CNN networks have demonstrated a significant difference between training and testing performance measures. The network accuracy during training reached 88.7% and 91.0% for AlexNet and VGGNet, respectively, while lower accuracy was demonstrated in testing. This could be caused by the limitation of the 2D CNN classification based only on RGB information with the lack of depth information. Moreover, this demonstrates that the network was not able to learn useful features to distinguish between different classes. To further investigate this thought, the model developed based on transfer learning from VGGNet was trained using five and four classes, where the classes related to the structures were grouped together. As VGGNet demonstrates a better performance, this model was selected as the network for a more detailed performance investigation. The combined classes represent more general object classes than the original seven class instances. To reduce the classes to five, the classes comprising of completely damaged and partially damaged were merged to create the class named damaged. Similarly, to reduce the total number of classes to a total of four classes, the classes completely damaged structures, partially damaged, and undamaged structures were combined to create a general class of structures. Identical parameters and architecture were used to train the new networks based on the reduced number of classes. It was observed that the training has improved in terms of accuracy, which turned out to be 92.0% and 94.6% for the five and four classes, respectively. Original confusion matrix of seven classes is shown in Figure 12. As for testing results, confusion matrices of merged five and four classes are shown in Figures 13 and 14, respectively. In the end, the VGGNet transfer learning using four classes has a significant improvement in both training accuracy and testing performance as expected. However, this model is not ideal for the targeted structural damage classification following natural hazards events. This is because the structural classes were combined, and the VGGNet training (in all models) cannot reliably distinguish between undamaged, partially damaged, and completely damaged structures. Instead, the general object classification of structure, roadway, terrain, and vehicles was proved to perform well. The improved performance when the classes combined demonstrate that depth information within 3D point clouds is critical to classify damaged structures from undamaged structures automatically.



**Figure 12.** VGGNet transfer learning confusion matrix of testing results on the Port Aransas dataset in seven classes.

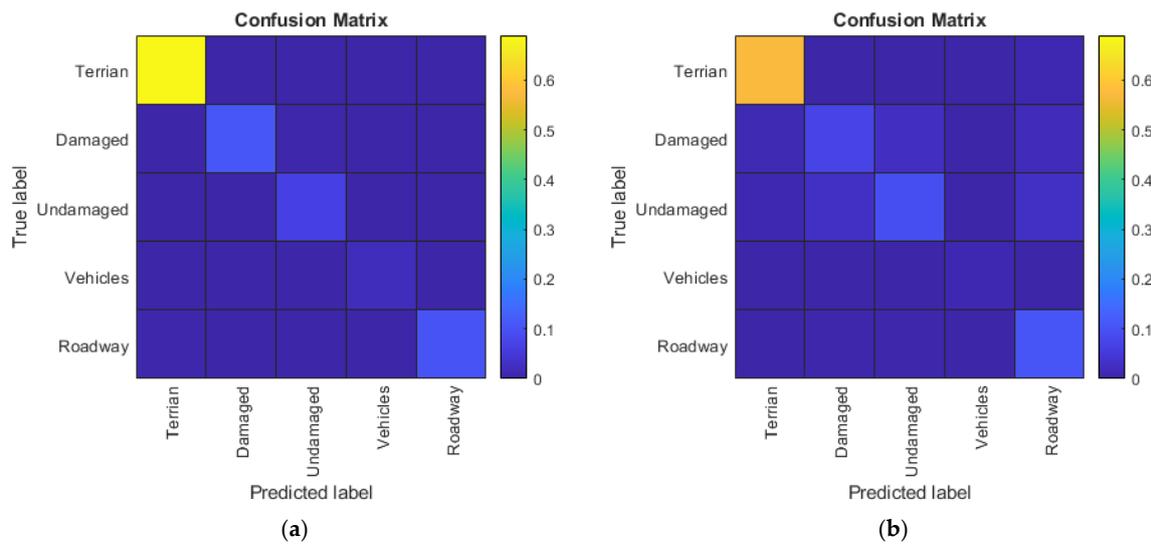


Figure 13. VGGNet transfer learning confusion matrix results for five classes: (a) training and (b) testing.

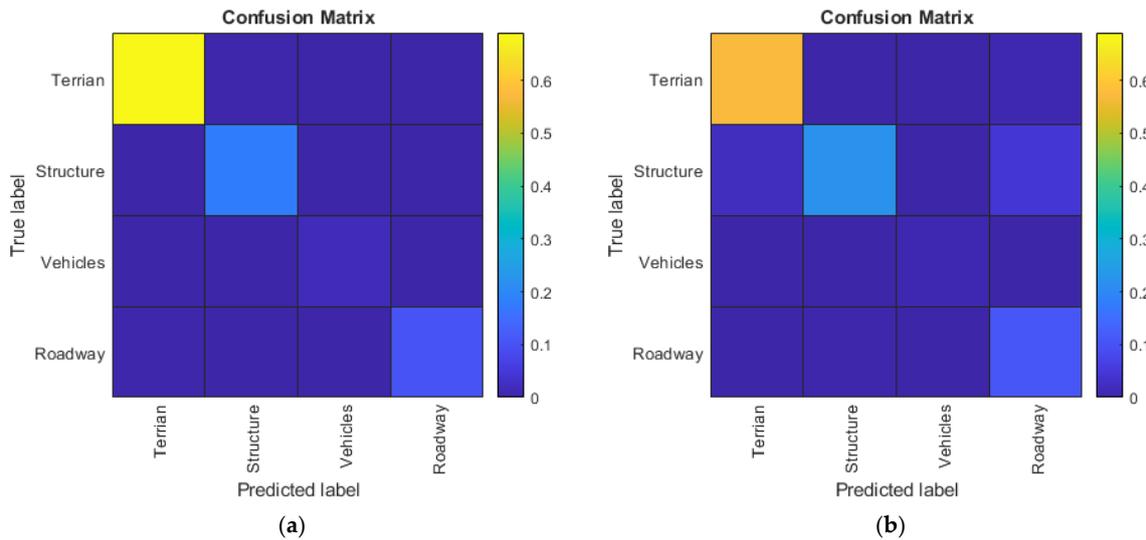


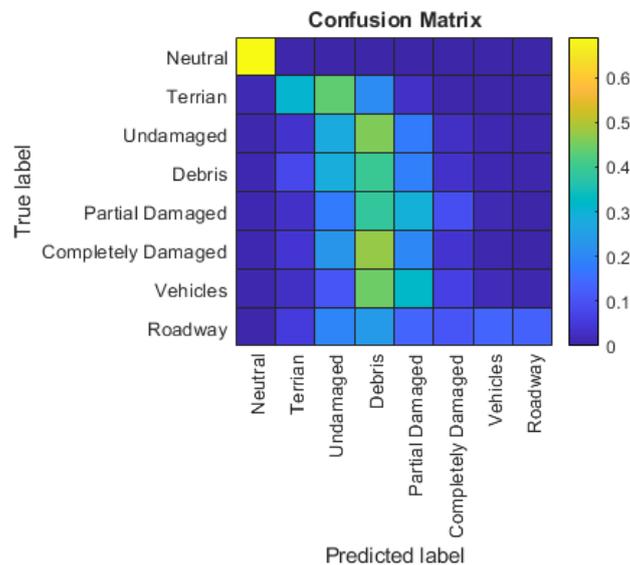
Figure 14. VGGNet transfer learning confusion matrix results for four classes: (a) training and (b) testing.

### 5.2. 3D FCN Experiment

Similar to the 2D CNN network, the 3D FCN model was developed and trained based on Salt Lake and Puerto Rico instances and was tested on the Port Aransas instances. To create the testing dataset, a procedure similar to that of creating the training dataset was followed; however, the testing instances were not augmented. Figure 15 shows the confusion matrix for testing on the Port Aransas dataset, and Table 5 provides the performance measures for each class.

The 3D FCN network prediction results on the test dataset demonstrated an overall similar performance measure in comparison to resulted performance measures observed during the training process. The overall cell accuracy of the network was 97.8%. The network was able to predict the class of terrain instances with a high level of accuracy while this was unexpected as the general terrain within the testing dataset in terms of texture and geometry differs in comparison to the training dataset. This suggests that the model was able to learn features that can generalize well between datasets with moderate to low similarities. A similar pattern in detecting the classes of partially damaged structures, completely damaged structures, debris, and vehicles to that of training was observed. Authors expect

that by performing extended training and using more learnable parameters, the network will learn features to distinguish between these classes with a higher level of accuracy.



**Figure 15.** 3D FCN confusion matrix for the Port Aransas dataset (in 7 classes).

**Table 5.** Quantified performance measures on the Port Aransas dataset.

Classes	3D FCN	
	Precision	Recall
Neutral	100	100
Terrain	61	31
Undamaged Structure	12	28
Debris	3	40
Partially Damaged Structure	10	29
Completely Damaged Structure	1	4
Vehicle	1	2
Roadway	95	13

### 5.3. Comparison of 2D CNN and 3D FCN

The detection accuracies of the 2D CNN models were consistently lower than the performance obtained based on the 3D FCN network, which is 91.0% for the 2D model and 97.8% for the 3D model. Comparing the structural damage detection performance, 3D FCN demonstrated a superior advantage over 2D models developed based on various class numbers. Keen advantages of 2D CNN and images relate to the smaller number of learnable parameters and reduced data sizes in comparison to the 3D FCN model. While 2D CNN performance improved from 92.0% to 94.6% in general object classification such as structures, terrain, roadway, or vehicles. This basic detection was not adequate for damage detection between structural classes like completely damaged, partially damaged, debris, and other classes. These results demonstrate a significant classification limitation when it is solely based on RGB information (which corresponds to 2D images) in comparison to RGB with depth information (which corresponds to 3D point clouds). Consequently, 3D FCN performs with a marked improvement in structural damage detection when comparing to 2D CNN.

## 6. Conclusions

Aerial image data collection provides an efficient technique to collect perishable data following a natural hazard event. Both 2D orthomosaic images and the 3D point clouds can be obtained and

processed for analyses and automated classification. This study compared post-event site damage classification using 2D and 3D datasets following two separate hurricanes from 2017 using 2D CNN and 3D FCN. The 2D CNN was conducted via transfer learning of two pre-trained networks: AlexNet and VGGNet. The inputs to the 2D CNN networks are 2D segmented images for unsupervised training and outputs the label for each image segment. The 3D FCN was conducted using aerial image-derived point clouds. Within the FCN method, point clouds are semantically classified into various classes. To keep the parameters consistent, both 2D CNN and 3D FCN have identical classes initially. To further demonstrate the 2D CNN classification performance, a reduction and combination of the classes were used for performance evaluation. The combination was aimed to eliminate the narrative of the classes of damage detection, which combines the structural damage and undamaged classes together.

Within the reduced numbers of classes for 2D CNN training, the accuracy improved at the cost of reducing and eliminating the classes corresponding to structural damage. The accuracy improvement demonstrates the 2D deep learning classifications are ideal for general object detection such as terrain, structures, vehicles, roadway, etc. However, they have a demonstrated limited learning capability to predict distinct structural characteristics from undamaged, partially damaged, and completely damaged as well as debris. However, this limitation was overcome when using a 3D point cloud dataset in deep learning, which contains both RGB and depth information. The model developed based on 2D data was only able to learn the dominant class (i.e., terrain) effectively. This results in lower precision and accuracy for other classes for both training and testing phases. On the contrary, the model developed based on 3D point clouds was able to learn other classes in addition to the dominant class. Classification within damage detection is known to be a class imbalance scenario, where the instances that represent damaged or debris are often the minority class that follows a random and unique geometric and color patterns.

Comparing the training durations, 2D CNN requires a significantly shorter time from a few hours to a day, while 3D FCN requires numerous days. The 2D CNN training accuracy achieved 88.7% and 91.0% for seven classes, and the highest accuracy was achieved by VGGNet training using four classes of 94.6%, while 3D FCN training accuracy was as high as 97.8%. However, when it comes to testing accuracy, the 2D CNN has significantly lower accuracy compared to 3D FCN. The accuracy decrease in the 2D dataset is expected due to the lack of depth information. Classification on 2D images is RGB based only, which can be influenced by the object surface reflection, sunlight, existence of shadows, etc. Moreover, despite 3D dataset preparation and network development is more time consuming, higher accuracy and reliability can be guaranteed. This is especially true when classifying the location and severity of damage following natural hazard events.

**Author Contributions:** Conceptualization, Y.L., M.E.M., and R.L.W.; data curation, Y.L. and M.E.M.; formal analysis, Y.L. and M.E.M.; funding acquisition, R.L.W.; methodology, Y.L. and M.E.M.; project administration, R.L.W.; supervision, R.L.W.; validation, Y.L., M.E.M., and R.L.W.; writing—original draft, Y.L., M.E.M., and R.L.W.; writing—review and editing, Y.L., M.E.M., and R.L.W. All authors have read and agreed to the published version of the manuscript.

**Funding:** No external funding directly supported this work.

**Acknowledgments:** This work was completed utilizing the Holland Computing Center of the University of Nebraska, which receives support from the Nebraska Research Initiative. Data related to the site in Texas were collected by Michael Starek of Texas A & M at Corpus Christi, and its availability is greatly appreciated by the authors, as published on the National Science Foundation's Natural Hazards Engineering Research Infrastructure (NSF-NHERI) DesignSafe cyberinfrastructure. Data related to Puerto Rico were collected by researchers under the supervision of Matt Waite of the University of Nebraska-Lincoln, and the sharing of this data with the authors is greatly appreciated.

**Conflicts of Interest:** The authors declare no conflict of interest. In addition, the funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

## References

1. Liao, Y.; Wood, R.L.; Mohammadi, M.E.; Hughes, P.J.; Womble, J.A. *Investigation of Rapid Remote Sensing Techniques for Forensic Wind Analyses*, 5th, ed.; American Association for Wind Engineering Workshop: Miami, FL, USA, 2018.
2. Adams, S.M.; Levitan, M.L.; Friedland, C.J. High resolution imagery collection utilizing unmanned aerial vehicles (UAVs) for post-disaster studies. In *Advances in Hurricane Engineering: Learning from Our Past*; American Society of Civil Engineers: Reston, VA, USA, 2013; pp. 777–793.
3. Chiu, W.K.; Ong, W.H.; Kuen, T.; Courtney, F. Large structures monitoring using unmanned aerial vehicles. *Procedia Eng.* **2017**, *188*, 415–423. [[CrossRef](#)]
4. Zhou, Z.; Gong, J.; Guo, M. Image-based 3D reconstruction for posthurricane residential building damage assessment. *J. Comput. Civil Eng.* **2016**, *30*, 04015015. [[CrossRef](#)]
5. Fernandez Galarreta, J.; Kerle, N.; Gerke, M. UAV-based urban structural damage assessment using object-based image analysis and semantic reasoning. *Nat. Hazards Earth Syst. Sci. Discuss.* **2014**, *2*, 5603–5645. [[CrossRef](#)]
6. Shin, H.-C.; Roth, H.R.; Gao, M.; Lu, L.; Xu, Z.; Nogues, I.; Yao, J.; Mollura, D.; Summers, R.M. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE Trans. Med. Imaging* **2016**, *35*, 1285–1298. [[CrossRef](#)] [[PubMed](#)]
7. Mohammadi, M.E.; Watson, D.P.; Wood, R.L. Deep Learning-Based Damage Detection from Aerial SfM Point Clouds. *Drones* **2019**, *3*, 68. [[CrossRef](#)]
8. Bengio, Y. Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*; Workshop and Conference Proceedings: Pittsburgh, PA, USA, 2012; pp. 17–36.
9. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In *Proceedings of Advances in Neural Information Processing Systems*; Communication of the ACM: Silicon Valley, CA, USA, 2017; pp. 1097–1105.
10. Berg, A.; Deng, J.; Fei-Fei, L. Large Scale Visual Recognition Challenge. 2010. Available online: <http://www.image-net.org/challenges/LSVRC/2010/> (accessed on 1 May 2010).
11. Oquab, M.; Bottou, L.; Laptev, I.; Sivic, J. Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, USA, 23–28 June 2014; pp. 1717–1724.
12. Hoskere, V.; Narazaki, Y.; Hoang, T.A.; Spencer, B.F., Jr. Towards automated post-earthquake inspections with deep learning-based condition-aware models. *arXiv* **2018**, arXiv:1809.09195.
13. Xu, Z.; Chen, Y.; Yang, F.; Chu, T.; Zhou, H. A Post-earthquake Multiple Scene Recognition Model Based on Classical SSD Method and Transfer Learning. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 238. [[CrossRef](#)]
14. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
15. Gao, Y.; Mosalam, K.M. Deep transfer learning for image-based structural damage recognition. *Comput. Aided Civil Infrastruct. Eng.* **2018**, *33*, 748–768. [[CrossRef](#)]
16. Olsen, M.J.; Kayen, R. Post-earthquake and tsunami 3D laser scanning forensic investigations. In *Forensic Engineering 2012: Gateway to a Safer Tomorrow*; Sixth Congress on Forensic Engineering: San Francisco, CA, USA, 2013; pp. 477–486.
17. Womble, J.A.; Wood, R.L.; Mohammadi, M.E. Multi-scale remote sensing of tornado effects. *Front. Built Environ.* **2018**, *4*, 66. [[CrossRef](#)]
18. Aixia, D.; Zongjin, M.; Shusong, H.; Xiaoqing, W. Building damage extraction from post-earthquake airborne LiDAR data. *Acta Geol. Sin. Engl. Ed.* **2016**, *90*, 1481–1489. [[CrossRef](#)]
19. Hackel, T.; Wegner, J.D.; Schindler, K. Fast Semantic Segmentation of 3d Point Clouds with Strongly Varying Density. *Int. Arch. Photogramm* **2016**, *3*, 177–184. [[CrossRef](#)]
20. Xing, X.-F.; Mostafavi, M.A.; Edwards, G.; Sabo, N. An improved automatic pointwise semantic segmentation of a 3D urban scene from mobile terrestrial and airborne lidar point clouds: a machine learning approach. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2019**, *4*.
21. Prokhorov, D. A convolutional learning system for object classification in 3-D lidar data. *IEEE Trans. Neural Netw.* **2010**, *21*, 858–863. [[CrossRef](#)]

22. Weng, J.; Luciw, M. Dually optimal neuronal layers: Lobe component analysis. *IEEE Trans. Auton. Ment. Dev.* **2009**, *1*, 68–85. [CrossRef]
23. Maturana, D.; Scherer, S. Voxnet: A 3d convolutional neural network for real-time object recognition. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 922–928.
24. Hackel, T.; Savinov, N.; Ladicky, L.; Wegner, J.D.; Schindler, K.; Pollefeys, M. Semantic3d. net: A new large-scale point cloud classification benchmark. *arXiv* **2017**, arXiv:1704.03847.
25. Zhang, F.; Guan, C.; Fang, J.; Bai, S.; Yang, R.; Torr, P.; Prisacariu, V. Instance segmentation of lidar point clouds. *ICRA Cited* **2020**, *4*.
26. Blake, E.S.; Zelinsky, D.A. *National Hurricane Center Tropical Cyclone Report: Hurricane Harvey*; National Hurricane Center, National Oceanographic and Atmospheric Association: Washington, DC, USA, 2018.
27. Smith, A.; Lott, N.; Houston, T.; Shein, K.; Crouch, J.; Enloe, J. *US Billion-Dollar Weather and Climate Disasters 1980–2018*; National Oceanic and Atmospheric Administration: Washington, DC, USA, 2018.
28. Pasch, R.J.; Penny, A.B.; Berg, R. *National Hurricane Center Tropical Cyclone Report: Hurricane Maria*; Tropical Cyclone Report AI152017; National Oceanic and Atmospheric Administration and the National Weather Service: Washington, DC, USA, 2018; pp. 1–48.
29. ASCE (American Society of Civil Engineers). Minimum design loads and associated criteria for buildings and other structures. ASCE standard ASCE/SEI 7–16. Available online: <https://ascelibrary.org/doi/book/10.1061/9780784414248> (accessed on 1 May 2019).
30. Beale, M.H.; Hagan, M.T.; Demuth, H.B. *Neural Network Toolbox™ User's Guide*; The MathWorks: Natick, MA, USA, 2010.
31. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
32. Ghazi, M.M.; Yanikoglu, B.; Aptoula, E. Plant identification using deep neural networks via optimization of transfer learning parameters. *Neurocomputing* **2017**, *235*, 228–235. [CrossRef]
33. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
34. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; pp. 234–241.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).