remote sensing

MDPI

Article

Technical Solution Discussion for Key Challenges of Operational Convolutional Neural Network-Based Building-Damage Assessment from Satellite Imagery: Perspective from Benchmark xBD Dataset

Jinhua Su [1], Yanbing Bai [1,*], Xingrui Wang [1], Dong Lu [2], Bo Zhao [3], Hanfang Yang [1], Erick Mas [4] and Shunichi Koshimura [4]

1 Center for Applied Statistics, School of Statistics, Renmin University of China, Beijing 100872, China; chasesu@ruc.edu.cn (J.S.); XingruiWang@ruc.edu.cn (X.W.); hyang@ruc.edu.cn (H.Y.)
2 The School of Computer Science and Engineering, The University of New South Wales, Sydney, NSW 2052, Australia; dong.lu@unswalumni.com
3 School of Informatics, The University of Edinburgh, Edinburgh EH8 9AB, UK; bo.zhao@ed.ac.uk
4 International Research Institute of Disaster Science, Tohoku University, Sendai 980-8572, Japan; mas@irides.tohoku.ac.jp (E.M.); koshimura@irides.tohoku.ac.jp (S.K.)
* Correspondence: ybbai@ruc.edu.cn; Tel.: +86-10-6251-1318

Received: 17 October 2020; Accepted: 16 November 2020; Published: 20 November 2020

Abstract: Earth Observation satellite imaging helps building diagnosis during a disaster. Several models are put forward on the xBD dataset, which can be divided into two levels: the building level and the pixel level. Models from two levels evolve into several versions that will be reviewed in this paper. There are four key challenges hindering researchers from moving forward on this task, and this paper tries to give technical solutions. First, metrics on different levels could not be compared directly. We put forward a fairer metric and give a method to convert between metrics of two levels. Secondly, drone images may be another important source, but drone data may have only a post-disaster image. This paper shows and compares methods of directly detecting and generating. Thirdly, the class imbalance is a typical feature of the xBD dataset and leads to a bad F1 score for minor damage and major damage. This paper provides four specific data resampling strategies, which are Main-Label Over-Sampling (MLOS), Discrimination After Cropping (DAC), Dilation of Area with Minority (DAM) and Synthetic Minority Over-Sampling Technique (SMOTE), as well as cost-sensitive re-weighting schemes. Fourthly, faster prediction meets the need for a real-time situation. This paper recommends three specific methods, feature-map subtraction, parameter sharing, and knowledge distillation. Finally, we developed our AI-driven Damage Diagnose Platform (ADDP). This paper introduces the structure of ADDP and technical details. Customized settings, interface preview, and upload and download satellite images are major services our platform provides.

Keywords: convolutional neural network; building-damage assessment; benchmark xBD dataset; disaster response online platform

1. Introduction

1.1. Motivation and Problem Statement

Natural disasters such as floods, hurricanes, or earthquakes cause great loss of life, property damage, and economic damage every year around the world. According to the World Health Organization, natural disasters kill around 90,000 people every year and affect close to 160 million

people worldwide [1]. When a natural disaster occurs, accurate information and effective responses are critical to saving thousands of lives and reducing loss. Knowing the location and severity of damages, emergency responders can respond quickly and deploy resources efficiently. Traditional workflows are based on ground-based assessments, which require a huge amount of labor and manual work for satellite imagery analysis [2–6], and are potentially impossible to obtain. For this, advanced automated methods are under development.

Although a series of research results has been achieved regarding convolutional neural network-based building-damage assessment from satellite imagery, there are still many challenges to be discussed. The release of large-scale xBD satellite disaster datasets [7] provides us with an excellent opportunity to discuss these issues. In this article, we combine our experience and experiments with the use of the xBD dataset to contribute to the following:

First, we made a comprehensive state-of-the-art review of convolutional neural network-based Building-Damage Assessment from Satellite Imagery.

Secondly, we conducted a technical discussion for the four key challenges of operational convolutional neural network-based building-damage assessment from satellite imagery as detailed below:

(1) Challenge 1: How Do We Objectively Compare the Accuracy of Various Methods in Case Evaluation Metrics Are Not Uniform?

(2) Challenge 2: How Do We Conduct Building-Damage Assessment in the Absence of Pre-Disaster Satellite Imagery?

(3) Challenge 3: How Do We Train a Robust Prediction Model Based on Disaster Data with Unbalanced Categories?

(4) Challenge 4: Which Technical Solutions Should Be Adopted to Improve the Accuracy of Building-Damage Evaluation Models?

Finally, we demonstrated the developed disaster emergency response platform: Cloud-Based AI Damage Mapping Online Service, a solution for realizing operational disaster damage assessment in the future.

1.2. xBD Benchmark Dataset

The data used in this paper is a wide-ranging satellite image dataset called xBD [8]. The xBD dataset is used by the xView2 prize challenge. As the largest building-damage assessment dataset to date, it contains both pre-disaster and post-disaster satellite images as well as 850,736 building annotations across 45,362 km^2 of land images.

Maxar/DigitalGlobal Open Data Program has high-resolution images in high definition from many sporadic regions of the world and images for xBD derived from this Open Data Program. 19 events as shown in Figure 1 are available in the complete xBD dataset across 22,068 images, and 850,736 building polygons are contained in the dataset. Each image has a 1024-by-1024-pixel resolution.

For pre-disaster imagery, a three-band RGB image and building polygons are provided. For post-disaster imagery, a three-band RGB image and building classifications based on The Joint Damage Scale are provided. A four-level granularity scheme is used to represent the damage levels of buildings from the images, namely no damage (0), minor damage (1), major damage (2), and destroyed (3). An example of the XBD dataset is shown in Figure 2.

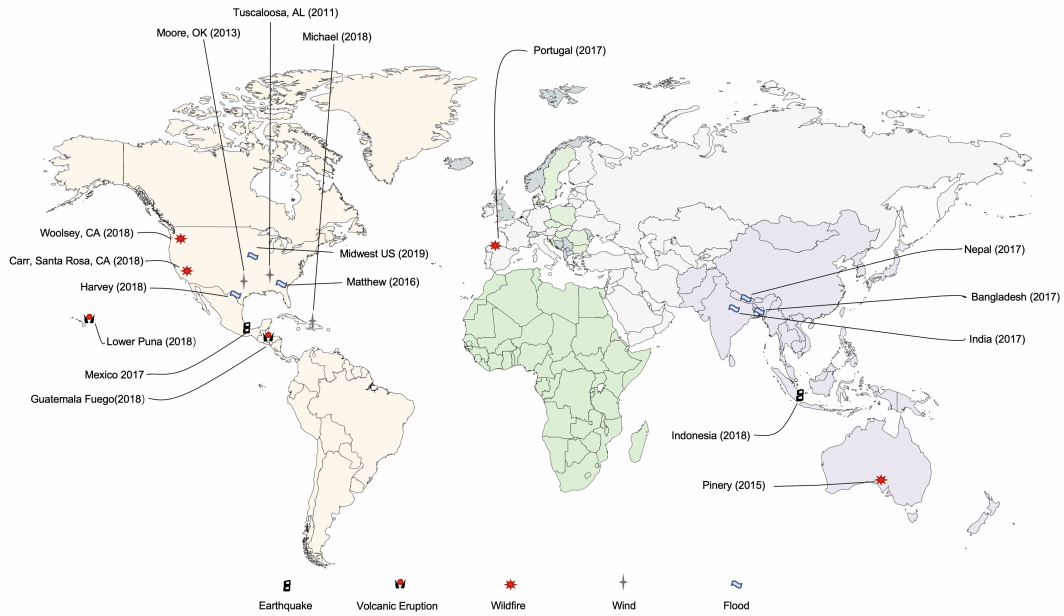


Figure 1. Disaster events included in the xBD dataset.

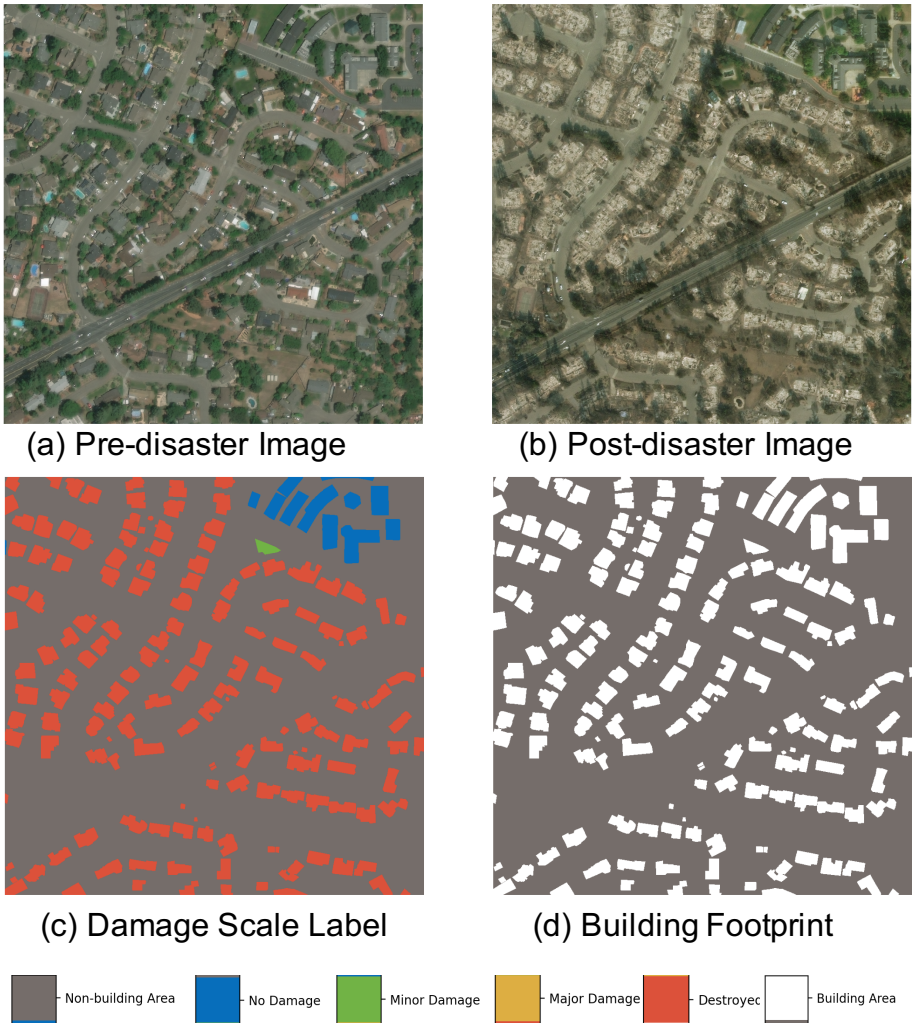
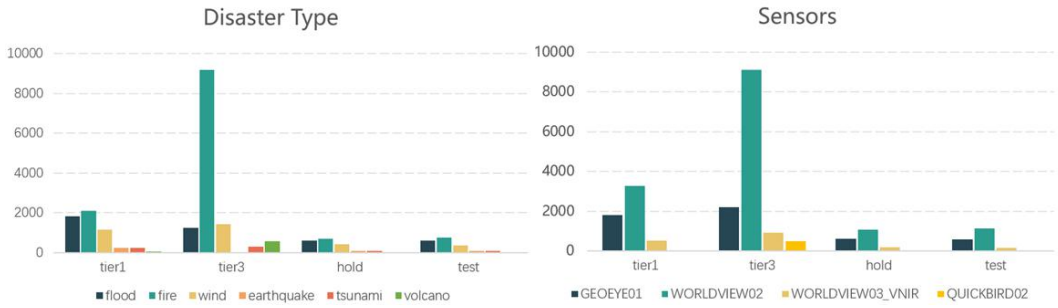


Figure 2. Example of xBD dataset. From left to right: (**a**) Pre-Disaster Image, (**b**) Post-Disaster Image, (**c**) Damage Scale Label, (**d**) Building Footprint.

Mining the group counting information in Figure 3, dataset tier 3 is different from the dataset tier 1, hold, and test, with more imagery of the wildfire and from the WORDVIEW-02 sensor. Different disaster types or sensor types differ in the imagery of the building damage.



(a) Group count by the disaster type (b) Group count by the sensor type

Figure 3. Histogram of (**a**) different disaster types and (**b**) different sensors images are obtained by in tier 1, tier 3, tier hold and tier test for xBD dataset. The distribution demonstrates the unbalanced distribution both in disaster types and sensor types in the dataset.

1.3. The Structure of the Article

Our paper is organized as per Figure 4. In Section 2, we review two categories of building-damage-level assessment models. In Section 3, we discuss four problems in applying intelligent damage-level assessment and propose some novel solutions. In Section 4, we develop a web-based platform supporting damage detection after disasters. Finally, we summarize the unexplored problems and further research objectives in Section 6.

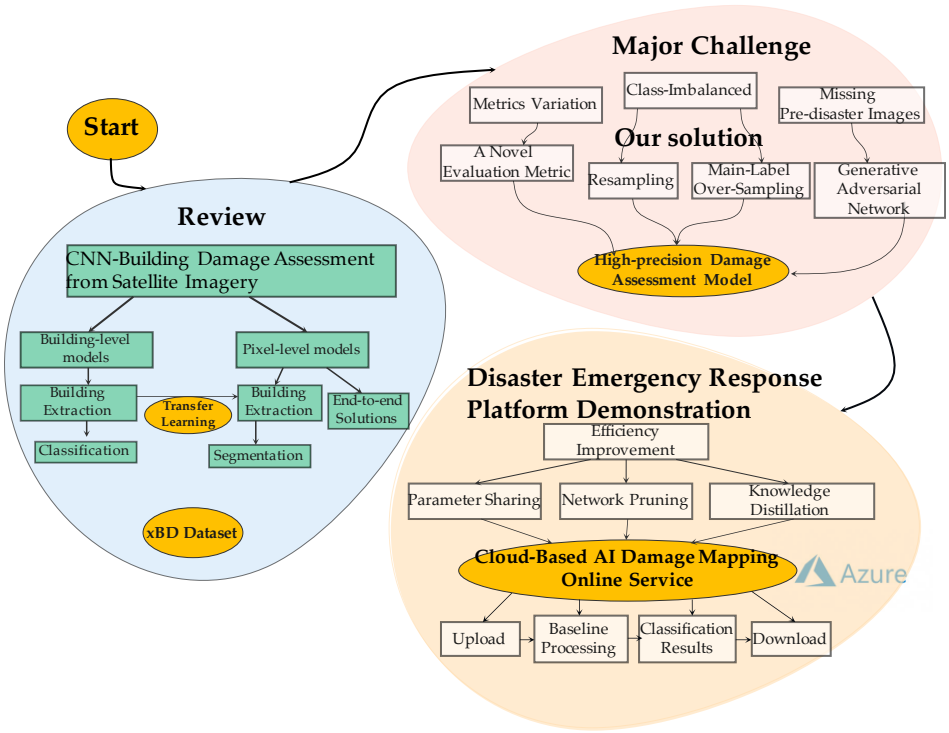


Figure 4. The overall structure of this paper. It contains three main parts and follows the black arrow line from the start point. The first is the review of two categories of building-damage assessment. The second is about four key challenges in this field and proposes some novel solutions. The third is the web-based platform supporting damage detection after disasters.

2. State-of-The-Art Review of Convolutional Neural Network-Based Building-Damage Assessment from Satellite Imagery

2.1. The Approaches of Damage Assessment at Building Level

For building-level tasks [7,9–14], each building has a ground-truth polygon that outlines its location in the image. It also has a ground-truth label that indicates the damage level of this building. As shown in Figure 5, the pipeline of building-level models is commonly composed of three main parts: a data preprocessing module, a two-class semantic segmentation module, and a four-class image classification module. The input, which can be a pre or post-disaster image, or a pair of pre-post images, will be entered into the semantic segmentation module after being preprocessed to get the buildings' location. The output from the previous module will be sent to the image classification module with its corresponding post-disaster image. Finally, the model will predict the damage levels of each building in the post-disaster image.

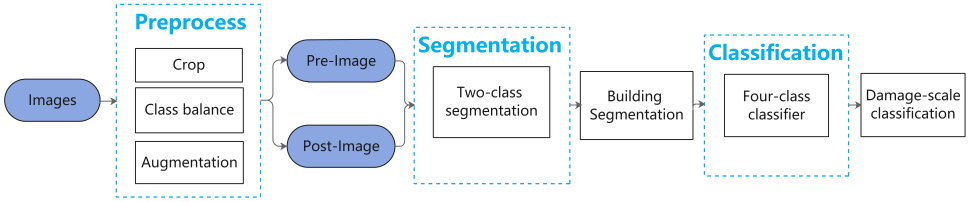


Figure 5. The flowchart of building-level model.

Data preprocessing is very useful to get a robust model and improve its performance. For example, we can use a crop to reduce memory usage and keep the image size consistent [15]. The crop is also very helpful if the position of the object has a large variance [16]. Sometimes we face the common imbalanced class problem. The resampling method is always used to limit the decrease in performance [17], which combines over- and downsampling that all classes contain the same amount of samples. Moreover, data augmentation methods, such as scaling, flipping, padding, and rotation, are applied during training to reduce overfitting [18]. The semantic segmentation part always uses the pre-trained model as its backbone [19]. It takes the image or a pair of images as the input and performs a pixel-wise binary semantic segmentation to predict whether each pixel belongs to a building or not. After that, the segmentation module can generate building masks by aggregating neighboring pixels, and hence we can get the locations of those buildings in the input images. The image classification part also uses some pre-trained models to be the core. Inputs to this module are the output polygons from the previous segmentation part and its corresponding post-disaster image. For a given polygon, a sub-image corresponding to the building can be cropped from the post-disaster image. Those sub-images, which have ground-truth labels that show the damage scales of those buildings, will be sent into the image classification module to get the predicted results.

The primary evaluation metrics for a building-level model are IoU (Intersection over Union), precision, recall, and F1 scores [20]. IoU is the intersection over union of the predicted bounding box and ground-truth box and used as the measure of loss. Precision (also called positive predictive value) is the fraction of relevant instances among the retrieved instances while recall (also known as sensitivity) is the fraction of the total amount of relevant instances that were actually retrieved. The F1 score conveys the balance between precision and recall. These metrics are used because train/test data are not balanced.

There are many prior works concerning the building-level method, exploring the effective use of deep learning in the context of damage assessment from pre- and post-disaster satellite images. xBD dataset is a felicitous dataset for training and validating damage-assessment models, and a two-stage building-level deep-learning network is proposed as the benchmark to assess the complexity of the xView2 challenge using xBD dataset [7]. In the baseline model, a SpaceNet that

features an altered U-Net structure [21] is used to extracted polygons indicating buildings from the original images, and a ResNet50 pre-trained on ImageNet [1] is used for damage-level classification of each building. Wheeler and Karimi [9] build a network with a rather similar structure to the baseline model of the xBD dataset, which also contains a localization module and a classification module. They test AlexNet, DenseNet, GoogLeNet, ResNet, and VGG for the classification module on a randomly selected subset from xBD, and find that ResNet enjoys the best performance on precision, recall, and F1 score, at the expense of relatively high time costs. Inspired by the xView2 challenge, Trevino et al. [10] incorporate the spatial properties of natural disaster damage into the process of assessing post-disaster damage of the buildings for a speedy and resource-efficient response operation. They design a hybrid GCN (Graph Convolutional Network) + CNN model and achieve a drastic boost over the xView2 baseline model.

Apart from xBD, several pieces of research are based on self-produced datasets. Fujita et al. [11] create the binary-class AIST Building Change Detection (ABCD) dataset, a combination of cropped building images from MLIT ("First report on an assessment of the damage caused by the Great East Japan earthquake". http://www.mlit.go.jp/common/000162533.pdf. (published in Japanese). Accessed: 2019-09-01) and PASCO (Corporation, P. http://www.pasco.co.jp/eng/. Accessed: 2019-09-01). They apply different methods to image cropping so that buildings can be extracted with a reasonable context. Later, three CNN structures, 6-channel, Siamese, and post-only are applied respectively to determine whether the building has been washed away, where the former two structures are used when both pre- and post-disaster images exist, and the last one is used when only post-disaster images are available. Xu et al. [12] also incorporate the localization task into the creation of the dataset. They collect images from three natural memorable disasters: the Haiti earthquake in 2010, the Mexico City earthquake in 2017, and 2018 Indonesia earthquake in 2018. Pre-disaster and post-disaster images are first fed into a building detection model to identify all buildings, where constructions in the regions are assessed as 'Severe Damage' and 'Destroyed' by UNOSAT [22] are labeled as positive, while buildings outside those regions are labeled as negative. They also test several different CNN structures, varying from single-stream models to double-stream models, for their performances on the classification task of buildings.

The advantage of using the building-level method is effectively using the homogeneous information in images and reducing the complexity of the task and removing the impact of image noise because it processes at building level instead of pixel level. However, it still has some drawbacks. The performance of the whole model largely depends on the performance of the first segmentation stage. The output of the segmentation module is a bounding box for a building instead of a pixel-wise mask, while the input of the classification module is the ground-truth label with the actual building, so it suffers some uncertainty for the prediction of the building location and maybe have poor performance than pixel-level method if the segmentation output is not accurate.

2.2. The Approaches of Damage Assessment at Pixel Level

2.2.1. The Idea of Pixel-Level Approach: Semantic Segmentation

Building-level models extract the buildings in the first place and then adopt some state-of-art classification networks to predict the damage scale label for the whole building with pre- and post-disaster images. Consequently, they often suffer loud noise under the uncertainty of building footprint predictions. The other stream of models, the pixel-level models, provide a new genre [20,23–26]. As Figure 6 shows, the current mainstream pixel-level approach is designed end-to-end instead of two-step. The input must contain a post-disaster image, which carries the source information for building-damage scale. Moreover, pre-disaster images are regularly combined to better mine the damage before and after the disaster. After the preprocessing series, which mainly includes cropping, ameliorating the class imbalance, and typical augmentation, the image is dropped in the pixel-level structure. The feature encoder and decoder process are adopted, and the output masks

are produced afterward, which contain c+1 channels: c damage-level labels plus one background label. Meanwhile, the building segmentation task and damage scale classification task of each pixel is accomplished based on the background label and c damage label, respectively. It should be noticed that the classification of pixel damage scale is valid only when its building segmentation result is true. In the training process, common classification loss functions such as cross-entropy loss, dice loss, focal loss, etc. are applied. Generally, the loss functions for building segmentation and damage scale classification are designed separately, and the weighted sum is taken as the total loss, which generally serves as the objective function. Metrics used to evaluate pixel-level models are similar to those of building-level models, including IoU, $F1_{loc}$, $F1_{cls}$, and so on.

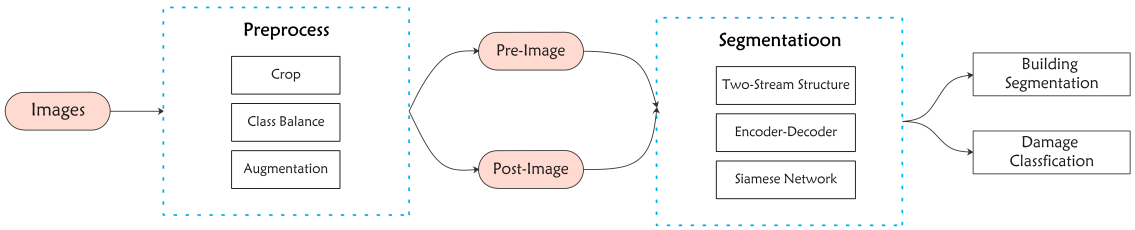


Figure 6. Pixel-level Computation Flow. The images are processed with crop, class balance and augmentation operations. Then, the processed pre-image and post-image are input to the encoder and decoder, and the masks of building segmentation and damage classification are produced.

The advantages of pixel-level models are self-evident. Compared with building-level models, a pixel-level model requires fewer parameters and training costs. However, there are some drawbacks of pixel-level models, the biggest of which is that different damage scale labels may appear in the same building. Thus, achieving high performance has become the focus of most existing research in the field of pixel-level damage assessment, and we will introduce two novel approaches in the following part of this section.

2.2.2. Innovative Solution: End-to-End Network

We refer to existing state-of-art pixel-level models. Hanxiang Hao et al. [25] design a Siam-U-Net-Attn model as Figure 7 shows. A double U-Net model will use both pre-disaster and post-disaster images to generate binary masks. In addition, features extracted by the encoder of the U-Net model will be used in the damage scale classification section. The two-stream features produced by the U-Net encoder and an independent decoder constitute the Siamese network, which compares features respectively extracted from two input frames to predict the damage levels of buildings. Weber et al. [26] use the Mask R-CNN (Regions with CNN features) architecture and apply the same model architecture for both the building localization task and the damage classification task. This way, the model can jointly reason about similar features. Instead of working with full images, they train the model on both pre- and post-disaster image quadrants and fuse the final segmentation layer to draw building boundaries more accurately.

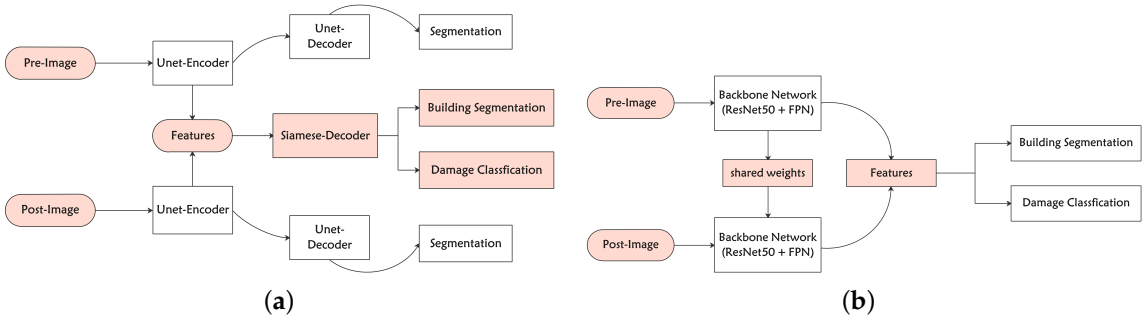


Figure 7. Examples of state-of-the-art end-to-end models. (**a**) The overall structure of the Siam-U-Net-Attn model [25] which uses a double U-Net model to generate binary masks. (**b**) The overall structure of the model in Weber et al. [26] which uses Mask R-CNN(Regions with CNN features) with FPN (Feature Pyramid Networks) architecture as the backbone.

2.2.3. Innovative Solution: Integration of Transfer Learning Ideas

The idea of transfer learning has been adopted in the current research. Transfer learning is a research problem in machine learning that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem [27]. Under the background of building damage, the first part of the building-level model, which has performed dramatically on building a segmentation task, can be transferred into the building-damage scale classification task. Specifically speaking, the primary output layer of a single-channel building segmentation model is replaced with the c+1 channel classification output layer structure as mentioned above, which makes full use of the high-precision building segmentation. As an example shown in Figure 8, Karoon Rashedi Nia and Greg Mori [20] propose a novel damage-assessment deep model for buildings using only post-disaster images. The model transfers three different neural networks: DilatedNet, LeNet, and VGG. VGG and LeNet extract deep features from the input source, and DilatedNet preprocesses the input data. Combinations of these networks are distributed among three separate feature streams. Then, the regressor summarizes the extracted features into a single continuous value denoting the destruction level.

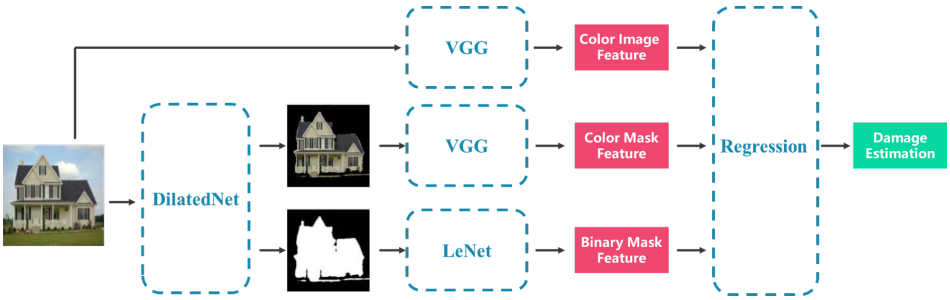


Figure 8. An example of transfer learning. First, we input the image to DilatedNet to obtain the color mask and binary mask. Then, the original image, color mask and binary mask are input to VGG, VGG and LeNet respectively for extracting corresponding features. Finally, we input three features to regression network and output the estimated damage.

3. Major Challenges and Our Solutions

There are some largely unexplored challenges in the application of intelligent building damage-level assessment. First, diverse metrics applied to evaluate the performances put an obstacle in comparing currently proposed networks. They are a building-scale network and a pixel-scale network. Secondly, drone images may be another important source apart from satellite images. However, pre-disaster drone images often lack historical data, which puts forward a demand for developing algorithms solely relying on post-disaster imagery. Thirdly, many networks built on the

xBD dataset suffer from poor classification performances in the 'Minor Damage' and 'Major Damage' class, primarily because of the limitations of satellite imaging methods and spatial resolution in detecting fine-scale target objects. In addition, data imbalance and the intrinsic ambiguity of the Joint Damage Scale also play an important role in precise classification. Fourthly, when applied to real-time rescue, a fast prediction is rather crucial. We will discuss the problems stated above precisely in this section and propose some novel solutions.

3.1. Challenge 1: How Do We Objectively Compare the Accuracy of Various Methods in Case Evaluation Metrics Are Not Uniform?

The diverse evaluation critics present an obstacle when comparing the performance of the current state-of-the-art models [7,12,28]. Normally, there are two specific evaluation ways for the damage-detection model. Some papers emphasize the pixel-level category accuracy of different damage status buildings [12,28]. To evaluate the pixel-level localization error and classification error, they linearly combine the average F1 scores of each class and IoU (intersect over union, which stands for the accuracy of localization task). However, the fixed hyper-parameter in the score function needs to be set manually, which cannot guarantee rational access to the model performance.

Some other papers care more about the distribution of building units [7]. These models first predict the region in the image containing buildings. Given the image region, the forward CNN classifier predicts the damage level of the buildings as the final result. The building-level detection seems a more robust method for detection in the real world since the rescue institution cares more about the localization point rather pixel accuracy, but since the accuracy of localization sometimes decides the accuracy of classification, the score of localization should not be ignored when comparing different models.

3.1.1. Solution 1: Conversion Between Two Metrics

Although these two methods of evaluation do not always correspond to each other since the pixel-level segmentation is more detailed than bounding box localization [29], and the pixel-level mask is considered more effective and detailed for the latter classification module, we still try to evaluate each of these methods by the other evaluation function. First, it is easy to convert the bounding box to a rectangle in pixels to apply the pixel-level metrics to a building-level prediction. There is one case where pixel-level model and building-level are evaluated at the pixel level. In Figure 9, the building-level model gives a more detailed building segmentation, while the pixel model outperformed in the damage classification. Secondly, it is not hard to apply the polygon processing tool from the Xview2 to generate the minimum bounding rectangle from the area of the same label, in order to get a building-level result from the pixel-level prediction. Both conversions cannot perform well, as the bounding box in building-level detection is too rough in pixel-level evaluation, and pixel-level segmentation is too sensitive to the input image, compared to the bounding box, which has a more robust location result.

3.1.2. Solution 2: Introduce a Novel Evaluation Metric

There has been some contribution in object detection tasks about finding an effective way to take both the precision of localization (pixel-level mask) and classification results into consideration. As we stated before, the damage-detection tasks which detect the location of the buildings and classify the level of damage status, to some extent, are similar to the multi-object detection tasks. A majority of the current object detection models [30–32] are designed to detect the possible location (bounding box) of the objects of interest and then classify the buildings inside the bounding box, for which the IoU score of the bounding box precision is a decisive factor. There is a sketch map to help judge the True Positive, False Positive, and False Negative of the building segmentation with the IoU, shown in Figure 10. An empirical but coarse method of prediction is to select all the bounding boxes with an IoU 0.5 and higher. Aiming to design a more comprehensive critic to define the performance of detection,

the COCO (Common Objects in Context) dataset [29] proposes to evaluate the classification result in different levels of IoU thresholds.

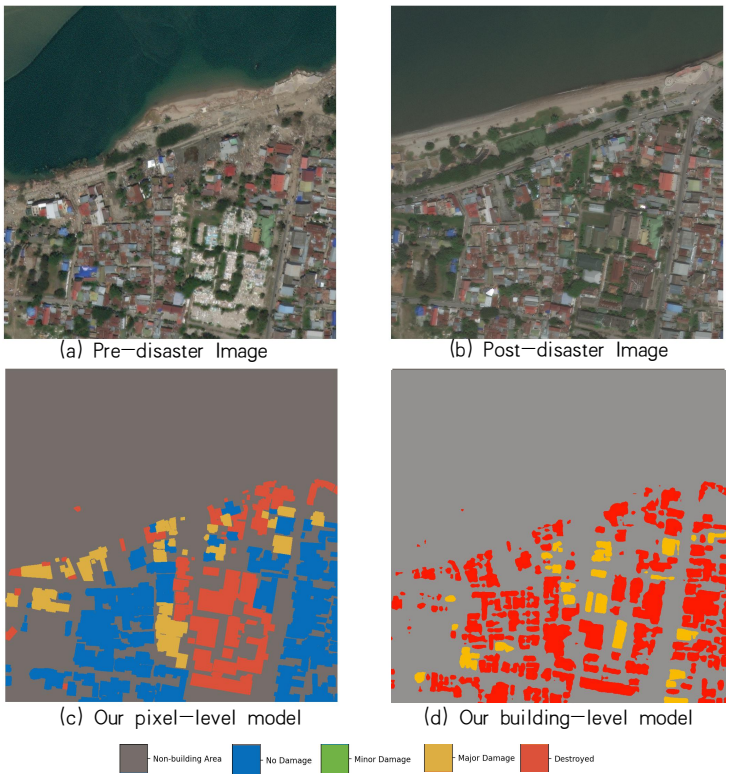


Figure 9. A case study of different level models evaluated in the pixel-level.

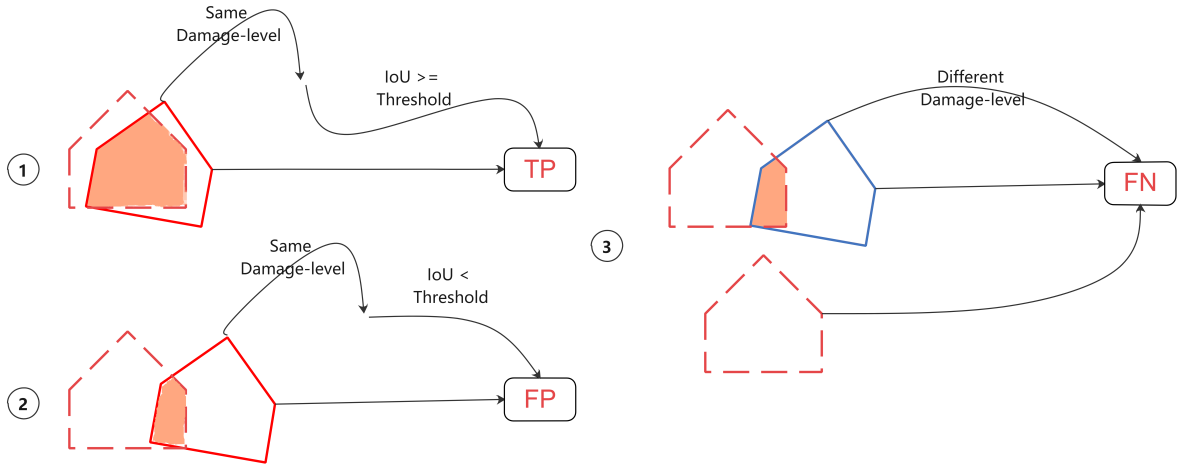


Figure 10. With a given IoU for building contour segmentation, we can define the True Positive (TP), False Positive (FP), and False Negative (FN) to evaluate the detection result. The IoU threshold decides the precision of localization result.

Inspired by the COCO dataset detection evaluation, we propose to evaluate the model F1 scores under different IoU thresholds, where the localization precision will be considered to be an intermediate result affected by the final classification, rather be generalized with classification accuracy by a fixed parameter [28], or ignored in object-level detection [7]. The object-level detection critics are special cases of F1 score matrix whose IoU threshold is equal to 0.5. Our metric, seen in Table 1, needs to calculate F1 scores with IoU equal to 0.5, 0.55, 0.60, ..., 0.90, 0.95, and the final result is the average of the above F1 scores with different IoU thresholds.

Table 1. F1 scores metrics under different IoU thresholds.

Metrics	Definition
mF1	Mean *F*1 score at IoU = 0.50:0.05:0.95
$F1^{IoU=0.50}$	F1 score at IoU = 0.50 (normal object-level metrics)
$F1^{IoU=0.75}$	F1 score at IoU = 0.75 (strict metric)

3.2. Challenge 2: How Do We Conduct Building-Damage Assessment in the Absence of Pre-Disaster Satellite Imagery?

The pre-disaster image and post-disaster image are crucial for the current state-of-the-art model to localize the building in the image. However, under the real application of damage-detection situations, it is common that the images are not recorded before the disaster.

3.2.1. Solution 1: Development of Building-Damage Assessment Methods Based on Only Post-Disaster Satellite Imagery

A trade-off way to overcome this setback is to train the localization module only by post-disaster image. For most damage-detection models, they use the temporal feature (pre- and post-disaster) and detect the difference of features between them. When losing pre-damage images, they localize the building and infer the damage level only from the post-damage image. This solution obviously will lose accuracy due to insufficient information.

3.2.2. Solution 2: Use of Generative Adversarial Network to Generate a Pre-disaster Image

It would be highly desirable to fill up the lost pre-disaster image-based on the post-disaster image. Fortunately, this task can be successfully achieved by Generative Adversarial Network (GAN) [33]. A generative model is designed to simulate the transfer function between distributions. GAN is a more recently proposed generative network that learns a loss and tries to classify if the output image is real or fake, while simultaneously training a generative model to minimize this loss [18]. It takes significant steps after convolution neural networks (CNNs) becoming the common workhorse for image processing. Concurrent works have applied GANs on image generation (generate images similar to the known datasets) [17] or image-to-image translation (generative images mapping from input images) [15,16,18].

Considering that the generative model is a relatively sophisticated technique in image generation tasks, we propose that these models can also be applied to our damage-detection tasks to generate a pre-disaster image of a building when only a post-disaster image is available. In our later experiment, we used PixelGAN [18] as our generative model. It "recovers" the damaged buildings in the post-disaster images and generates the paired pre-disaster ones. We compare the real post-disaster image to our generated image and show that PixelGAN achieves a reliable pre-disaster image generation. Furthermore, we exploit the generated pre-disaster images and compare them with the detection result and with the model using a real pre-disaster image and the result that only uses post-disaster images. The result shows that the generated image can improve the result significantly when the pre-disaster image is not available.

3.3. Challenge 3: How Do We Train a Robust Prediction Model Based on Disaster Data with Unbalanced Categories?

If there is an unequal distribution between its minority and majority class, a dataset can be referred to as imbalanced. Data imbalance has long been a problem in data analysis and prediction. A model is likely to classify most instances as the majority class to yield a fairly high metric, even though the performance of the minority class is often of the most significance. Multi-label imbalance occurs when data distributes unevenly across multiple classes (i.e., more than two classes) and plagues the performance of a classifier on the class with much fewer instances than the others. For networks built

on the xBD dataset, the performances on instances labeled as "Minor Damage" or "Major Damage" are often unsatisfactory, primarily because they account for only a small part of the whole dataset, shown in Table 2.

Table 2. Non-building Area and building area ratio in pixel level.

Non-Building	No Damage	Minor Damage	Major Damage	Destroyed
96.97%	2.33%	0.24%	0.27%	0.18%

Several methods have been proposed to address the problem of data imbalance, which mainly fall into two categories: data-level methods and algorithm-level methods [34]. The former includes data sampling and feature selection, while the latter consists of cost-sensitive, ensemble, and hybrid measures. In this section, we will introduce some techniques, including both data-level and algorithm-level ones, to address the poor performance of the "Minor Damage" and "Major Damage" instances. We will also discuss some problems worth considering when the data imbalance within the current four classes on the xBD dataset is well solved.

3.3.1. Solution 1: Data Resampling Strategies

Data sampling includes over-sampling and under-sampling. The former adds instances of the minority class while the latter removes instances of the majority class. Random over-sampling (ROS) and random under-sampling (RUS) can address data imbalance effectively despite their simplicity [35]. However, the problem with our task is more complicated. The proposed network conducts pixel-level classification with input data in the form of pixel-labeling images, and nearly all images in the dataset contain pixels of different classes. Consequently, the sampling process needs to be adjusted to address the mismatch between the classification unit and the input unit. In this paper, we introduce 3 specific methods: Main-Label Over-Sampling (MLOS), Discrimination After Cropping (DAC) and Dilation of Area with Minority (DAM). We also apply the Synthetic Minority Over-Sampling Technique (SMOTE) [36] to generate some fake images of minority classes.

- Main-Label Over-Sampling (MLOS)

MLOS can be regarded as an over-sampling method at the image level. To determine how many times an image, which covers pixels of both majority and minority classes, should be repeated in the final training dataset, we will first introduce the concept of main label. If $\boldsymbol{n}^{(i)} = [n_0^{(i)}, n_1^{(i)}, n_2^{(i)}, n_3^{(i)}]^T$ is the vector recording the number of pixels of each class in image i, and $\boldsymbol{w} = [w_0, w_1, w_2, w_3]^T$ represents the relative importance between different classes, the main label of image i (denoted as $L^{(i)}$) is determined as follows:

$$L^{(i)} = \arg\max_j \ n_j^{(i)} w_j, \ \ j = 0, 1, 2, 3, \tag{1}$$

where class 0 denotes "No Damage", class 1 denotes "Minor Damage", and so on. A function $\mathcal{F}(L^{(i)})$ will then be applied to decide how many times the image i will be repeated. The weight vector w and the function $\mathcal{F}(L^{(i)})$ are flexible, changing along with the structure of the dataset, the mechanism of the network, and perhaps the specific training environment. Generally, vector $\boldsymbol{w}$ will emphasize the minority classes, and $\mathcal{F}(L^{(i)})$ will allow images main-labeled as the minority class to be repeated more than the ones main-labeled as the majority class.

- Discrimination After Cropping (DAC)

DAC is applied after MLOS. The original image size in the xBD dataset is 1024×1024. We uniformly sample smaller crops, e.g., 512×512 as the input of the network. We would sample

several crops from an image, re-weight each pixel with vector $\boldsymbol{w'}$ inversely proportional to the frequencies of damage levels, and choose the crop with the largest weighted sum $\boldsymbol{ws}$. Similar to MLOS, the number of crops sampled from one image and the vector $\boldsymbol{w'}$ are changeable. DAC combines the idea of over-sampling and under-sampling since it removes crops with more majority class pixels and adds the ones with more minority class pixels simultaneously. Therefore, DAC can alter data distribution without increasing the data volume.

- Dilation of Area with Minority (DAM)

After the implementation of DAC, we will further balance the numbers of pixels of different classes. DAM is introduced to alter the distribution of damage levels within the same image. We expand areas with pixels labeled as the relative minority classes, and pixels in the overlapping regions are re-labeled as the minor class. For instance, if the dilated area of class *Minor Damage* overlaps the dilated area of class *Destroyed*, pixels in the overlapping region will be uniformly labeled as *Minor Damage*, on which the network prediction is worse in the metric of F1. DAM is a sampling method at the pixel level, and it also addresses data imbalance without increasing the data volume.

- Synthetic Minority Over-Sampling Technique (SMOTE)

We also implement the classical data-sampling method SMOTE to generate some fake images. In our task, we swap pixel locations in images *main-label* as minority classes for some new images. Such an operation can also diversify the spatial patterns of damage-level distribution and thus enhance the generalization ability of our model. It should be noted that there are various approaches to synthesize images featuring minority classes, and the location swap is only one of them. More sophisticated methods, such as GAN, can also be applied in this process.

3.3.2. Solution 2: Cost-Sensitive Re-Weighting Schemes

Algorithm-level methods seek to address data imbalance by altering the objective function or the model structure. Cao et al. [37] design a wrapper framework incorporating classification performance metrics, such as area under curve (AUC), directly into the objective function of the SVM model. In this paper, we apply a similar mechanism to improve the performance of our network on instances of minority damage levels, introducing a loss function combining IoU and focal loss.

It should be noted that the choice of metrics is crucial when data is imbalanced. Some metrics, such as accuracy and error rate, can be misleading since the model is likely to classify most of the instances belonging to the majority class for a rather high score, although the performance of the minority class may be more important. Therefore, we combine two metrics sensitive to data imbalance, dice and focal loss, for a compounded loss function. Dice is a measure to evaluate the similarity between two groups and is equivalent to the F1 score when applied in the classification task. We apply a macro dice score for the multi-classification task, which can be represented as follows:

$$Dice_j = \frac{2TP_j}{2TP_j + FN_j + FT_j}, \quad j = 0, 1, 2, 3 \tag{2}$$

$$Dice = \frac{1}{4}\sum_{j=0}^{3} Dice_j, \tag{3}$$

where TP_j denotes the number of pixels correctly classified as category j, FP_j denotes the number of pixels misclassified as category j, and FN_j represents the number of pixels inaccurately classified as other categories.

Focal Loss is another well-known metric concerning classification of imbalance data, which is based on cross-entropy with the parameter γ to adjust the importance of ambiguous instances and α to

adjust the relative importance of the minority class. In the multi-classification task, the focal loss can be represented as follows:

$$Focal\ Loss = -(1 - p_{prediction} \times p_{groundtruth})^{\gamma} \log(p_{prediction}), \quad (4)$$

where $p_{groundtruth}$ denotes the frequency of the ground-truth damage level and $p_{prediction}$ is the possibility given by the classifier of the current pixel belonging to its ground-truth class.

We combine the values of dice and focal loss for the final compound loss function, which is calculated as follows:

$$Loss = \beta \times Dice + (1 - \beta) \times Focal\ Loss, \quad (5)$$

where the parameter β represents the relative importance of these two metrics.

There is one case showing the improvement of applying different techniques to tackle the category imbalance problem based on our pixel-level model, PPM-SSNet (Table 3). As for the naïve model, the model will perform well in the majority category with the worst result for the other categories. With data resampling, our model improves a lot in the destroyed class. With weighted loss added, the model slightly improved the performance in the minor and major damage classes.

Table 3. Ablation study of category imbalance technique (%).

Methods	P_{clf_1}	R_{clf_1}	$F1_{clf_1}$	P_{clf_2}	R_{clf_2}	$F1_{clf_2}$	P_{clf_3}	R_{clf_3}	$F1_{clf_3}$	P_{clf_4}	R_{clf_4}	$F1_{clf_4}$	$F1_{clf}$
Naïve PPM-SSNet	90.81	94.12	92.43	15.75	32.01	21.12	30.23	37.54	33.43	72.41	31.23	43.61	36.01
+ Data Resampling	96.04	67.93	79.51	20.69	73.64	32.28	58.28	70.12	63.69	80.49	74.17	77.25	55.41
+ Data Resampling + Weighted Loss	90.64	89.07	89.85	35.51	49.50	41.36	65.80	64.93	65.36	87.08	57.89	69.55	61.55

3.3.3. Rethinking: Continuous Label Problems about Data

Despite various methods mentioned above to address data imbalance, there are still some problems concerning our task that remain unsolved. We build the network on the xBD dataset, a four-class dataset about building-damage assessment after environmental disasters. xBD applies the Joint Damage Measurement varying from no damage (class 0) to destroyed (class 3), which is a trade-off between the precision and the convenience of annotation [7]. Unfortunately, the classification of building damage is quite different from the classification of different species such as cats and dogs, and the Joint Damage Scale itself is subjective and sometimes indistinguishable between adjacent classes. In other words, our proposed network is trying to learn a classification rule with intrinsic ambiguous boundaries. Lacking damage details to distinguish them from satellite imagery may contribute to this problem.

Other problems are less relevant to data imbalance but still plague the reliability and universality of our network. In practical application scenarios, images taken by drones may be used as input because of the unavailability of satellite images. Drones tend to join multiple small images of buildings for a larger one to compensate for its relatively small spectrum compared to the satellite. As a result, the input images are no longer in rectangular form, and their irregular boundaries might be misleading to the classification task after convolution operation. Another problem also concerns the unfavorable input. It is possible that we do not have access to pre-disaster images immediately after disasters when humanitarian assistance is urgent. We have discussed some possible solutions to this problem in the precedent text.

3.4. Challenge 4: Which Technical Solutions Should Be Adopted to Improve the Efficiency of Building-Damage Evaluation Models?

Due to the demand for rapid building-damage assessment in the case of humanitarian assistance and disaster recovery research after environmental disasters, it is crucial to modify the neural network

for faster prediction. In this section, we enhance the efficiency of prediction by altering data-processing methods and compacting the neural network.

3.4.1. Solution 1: Feature-Map Subtraction

To figure out the damage level, the network would combine the feature maps of each pair of pre- and post-disaster images. Generally speaking, we can either concatenate the feature maps or conduct certain calculations on them to produce a comprehensive one. Despite the favorable performance the concatenation method may produce, this operation would increase the number of coefficients in the network along with the expanding image size. Therefore, we choose to apply certain calculations for a comprehensive feature map. Since difference detection between pre- and post-images is crucial in damage-level assessment, we subtract the post-disaster feature map from the pre-disaster one. It should be noted that subtraction is not the only operation that can be conducted on feature maps. More sophisticated methods are welcomed as long as they produce better performance without significantly aggravating the computational burden.

3.4.2. Solution 2: Parameter Sharing

We implement the structure of Siamese Networks [8] to further reduce computational expenses. Instead of training independent encoders for pre-disaster images and post-disaster images, respectively, we modify the network so that the front parts of encoders can share weights. Since the mechanism of encoding pre-disaster images and post-disaster images are nearly identical, such a structure is reasonable. This weight-sharing mechanism would reduce the number of parameters in the network thus accelerating the training and predicting process.

3.4.3. Solution 3: Knowledge Distillation

Knowledge distillation aims to achieve a light network with considerable accuracy. Formally proposed by Hinton et al. [38] in 2015, it is presently applied in many fields, including image classification [39], biometric identification [40], object detection [41], and semantic segmentation [42]. Knowledge distillation would transfer knowledge acquired in the teacher net, which is heavy but has high accuracy, to the student net, which is light but has low accuracy [43]. In this way, knowledge distillation considerably saves computational expense without exacerbating the performance of the network. In this paper, we develop a loss function, which includes knowledge distillation losses for both features and outputs of the networks and train a relatively lighter student network from a pre-trained teacher network. The algorithm is Algorithm 1.

Algorithm 1: Knowledge Distillation for Satellite Image Segmentation

Input: $(\mathbf{x}_i, y_i)$: training data, $i = 1 \cdots N.$, $\phi_{\cdot T}$ and $\phi_{\cdot S}$: Pre-trained teacher and student model.

while *not converge* **do**
  for $i = 1, \cdots, N$ **do**
    $\mathbf{f}_i^T, \mathbf{o}_i^T = \phi_{\cdot T}(\mathbf{x}_i)$ // Compute features and outputs produced by teacher model.
    $\mathbf{f}_i^S, \mathbf{o}_i^S = \phi_{\cdot S}(\mathbf{x}_i)$ // Compute features and outputs produced by student model.
    $L_{cls} = -\log \frac{e^{\mathbf{o}_{i,y_i}^S}}{\Sigma_{k=1}^K e^{\mathbf{o}_{i,k}^S}}$ // Compute classification loss. K is the number of classification categories.
    $L_{kf} = \|\mathbf{f}_i^S - \mathbf{f}_i^T\|_F^2$ // Compute knowledge distillation loss for features.
    $L_{ko} = -\Sigma_{k=1}^K \frac{e^{\mathbf{o}_{i,k}^T/H}}{\Sigma_{k=1}^K e^{\mathbf{o}_{i,k}^T/H}} \cdot \log \frac{e^{\mathbf{o}_{i,k}^S/H}}{\Sigma_{k=1}^K e^{\mathbf{o}_{i,k}^S/H}}$ // Compute knowledge distillation loss for outputs. H is the hyper-parameter for obtaining distilled outputs. Usually, we set $H = 2$.
    $L = L_{cls} + \alpha \cdot L_{kf} + \beta \cdot L_{ko}$ // The final loss. α and β are hyper-parameters.
  end for
end while

Output: $\phi_{\cdot S}$

There is a case study about to what degree the knowledge distillation helps the student model reduce the parameters and achieve heavy model performance. We conduct some experiments based on the first-place model in the xView2 Challenge whose the source has been published on the GitHub website. By making each layer about half the channels of the previous layer, we name it as the student model. In Table 4, the student model can achieve nearly the same results in the building localization task, with about 91% capability of the teacher model in the damage classification task.

Table 4. A case study of the knowledge distillation.

Models	IoU for Building Localization	Overall F1 Score for xView2 Challenge
Teacher (xView2 1st place model)	0.84	0.79
Student (about half parameters of Teacher)	0.82	0.72

3.4.4. Solution 4: Network Pruning

Apart from the methods introduced above, there are many other ways to compact the network and accelerate the prediction, one of which is network pruning. Network pruning, which removes redundant weights and only preserves the important ones, proves to be an effective technique to improve the efficiency of deep networks when computational resources are limited. A typical weight-pruning process includes three stages: training a model with an excessive number of parameters, pruning the model based on certain criteria, and fine-tuning the pruned model to regain the loss performance [44]. In building-damage assessments, weight-pruning can be applied to make predictions faster since the pruned network contains much fewer coefficients. Such work in building-damage assessment based on satellite images remains unexplored but is well worth trying.

4. Results: Disaster Emergency Response Platform Building Challenges: Cloud-Based AI Damage-Mapping Online Service

AI-driven Damage Diagnosis Platform is a web-based platform to support the detection of building damage in a disaster by accessing and visualizing the remote-sensing images pre- and post-disaster. The system allows the visualization of images, both pre-disaster and post-disaster, as well as the damage-detection result. The platform offers a set of basic functionalities such as sliders and zoom-in tools. Our platform is available at http://qwenwu.online/classify/public/index.html for a period.The platform interface is shown in Figure 11.

When a disaster occurs, quick response reinforcement needs accurate and instant data to call out situation awareness and implement effective countermeasures. This multi-level web application is a cloud platform that consists of a graphic interface and smart algorithm buttress, dedicated to highlighting damaged buildings after disasters and assessing the extent of damage, hence classifying the buildings of interest into four appreciated categories, “no damage”, “minor damage”, “major damage” and “destroyed”. These labels are called Joint Damage Scales, an integrated evaluation scale for construction damage in satellite imagery primarily based on Damage-Assessment scales proposed by HAZUS, FEMA, and Kelman.

To build an AI platform to present the latest model of the field of building-damage detection, which has not been widely applied in the industry, helps the peers and the demand side to promote this technique into the application environment. There are four challenges for a data science team to tackle. In this paper, we give our solutions to these challenges according to the case of our platform.

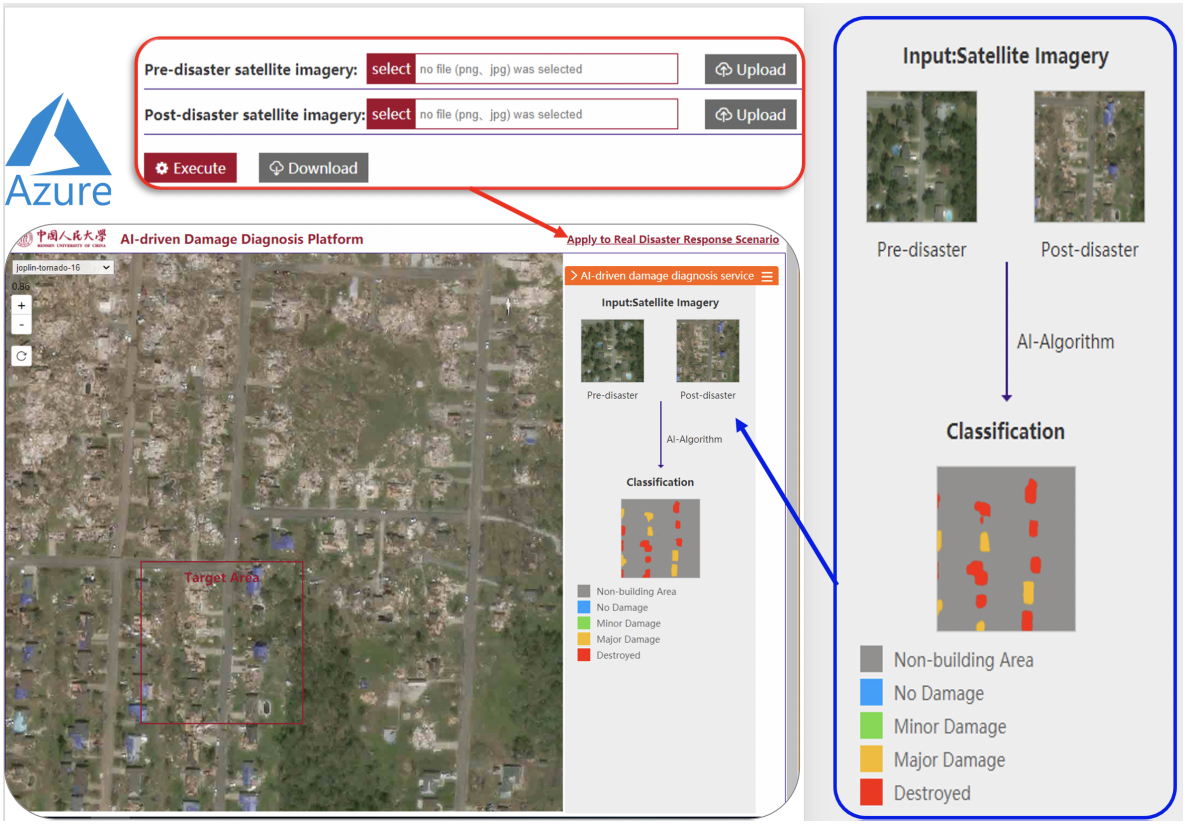


Figure 11. Interface of Cloud-Based AI Damage Mapping Online Service.

4.1. Challenge 1: How to Continuously Give State-of-the-Art Prediction?

The major motivation to build our platform is to present the potential application value of state-of-the-art building-damage detection techniques. The research in this field has drawn the attention of many computer vision researchers, and evermore efficient models have been put forward. This platform needs to update the model to give state-of-the-art predictions. Currently, our running model is PPM-SSNet, a Pyramid Pooling Module-based Semi-Siamese Network for End-to-End Building-Damage Assessment from Satellite Imagery, which will be published soon.

4.1.1. Splitting the Whole Procedure into Several Minima Execute Units

Our platform provides the service of assessing the input images and giving the damage-level images, which includes three main procedures. First, the data-processing unit is to regulate the image scale and format for prediction. The preview interface will cut the demo images saved in the server previously, while the upload interface will transform the image format and check the consistency of the pre-disaster image and the post-disaster image. Second, the prediction unit is to get semantic segmentation results by the model. Third, the get-results unit is designed for the user-end browser to take results that have been saved in the server. The prediction unit and get-results unit will be run automatically after the data-processing unit in the preview interface, while the two units can be manually controlled in the upload interface.

Every unit will return at least one response result for the web page and is suitable to be wrapped in the API. As Table 5 shows, we offer six different APIs for users to upload images and obtain their expected damage scale classification masks. Specifically, the user uploads their images on the web in the first place via API "upload" or "cls for upload". Afterward, "get cls image" will return the results of the classification. The user can check or download the results using "download" or "check download result" API, respectively.

Table 5. API description table.

API	Request Type	Usage	Parameter	Result
get-sub-image	post	cut and get the region	id (image number) x (left-top x coordinate) y (left-top y coordinate)	id pre_cut (cut pre- image), post_cut (cut post- image)
get-cls-image	get	get the classification result	id	img (classification result)
upload	post	upload image	file type (pre- or post-damage), fileName (filename uploaded)	fileName (uploaded file name)
download	get	download image	fileName	file
cls-for-upload	post	managing pre-and-post images	preName, postName	fileName (the result file name)
check download result	get	check if the manage process is done	fileName	is Finish (false/true)

4.1.2. Making the Prediction Unit a Highly Changeable Box

The prediction unit will be updated in the future and is a unique black box for other APIs. We deployed two series of mainstream models to tackle damage-assessment and classification problems, whose processing grains are building-level and pixel-level. For building-level models, there is the building localization part and the damage classification part, which can be taken as the two-class semantic segmentation task and four-class image classification task. It is easier to apply the pixel-level model since it is just one task of the five-class semantic segmentation. The building-level result can be converted into pixel-level results by padding each pixel of one polygon with the same prediction label.

Unlike the procedure that differs between the building-level and the pixel-level, the difference of the model structure is more common. Indeed, any semantic segmentation model can be converted into the Siamese network structure to process the image before and after the disaster to get the change-detection results. In recent years, more and more network structures have been put forward. As long as experiments prove that new semantic segmentation achieves state-of-the-art assessment of the building-damage task, it is easy to update the model of this platform.

4.2. Challenge 2: How to Meet the Need for Both the Visitors and the Real Demand Side?

Building a platform is technically hard but worthwhile for researchers to present their work, which makes it easier to explain the potential application value. The users of this platform are two-fold. On the one side, a stable API, which provides an available baseline for some contrast experiments or case studies, is urgently needed. On the other side, it is unnecessary to know about the details of the AI algorithm for visitors from the industry. The design of a simple and friendly user interface, as well as a flexible and stable API, meets the needs of both the visitors and the real demand side.

4.2.1. Demo Image and Friendly Interface Design for the Visitors

The platform gives a quick demo interface for visitors to know about what this application can do at one glance. Demo images on the platform are ground-truth satellite images, unbiased overhead views, including several images taken before and after the Joplin tornado and Santa Rosa wildfire. Images in the Joplin demo resembled the extent to which the typhoon had affected the dense residential areas. A massive number of dwellings are destroyed with only their sites left. On the contrary, images in the Santa Rosa demo depicted how sparse residential areas are affected by wildfire. In particular, we choose the images from the worst-hit area where the diversity of damage degree can be assured and present a better demonstration for classification. It should be noted that all input images of the algorithm are of the same size due to a size-fixed selection area following the user cursor in the image. In addition, this limitation, in some ways, precluded the model from processing images

with incongruous sizes. To adjust the viewable range in the selection area, one must zoom in or zoom out the original image by clicking the buttons with plus and minus symbols. The visualization panel has three parts—one part to present the original post-disaster image in the selection area, one part of presenting the corresponding pre-disaster image, and a part in which buildings are highlighted in colors consistent with damage degree. When user customization is imported on labels, you will see a smaller panel representing the legend of damage degree, and other functions enable users to customize their appreciated groups with certain colors and basic categories as components.

4.2.2. Image Upload and Download API for the Real Demand Side

To improve the application ability of the platform in comprehensive scenarios such as tsunami, flood, volcano eruption, wildfire, earthquake, and so on, an interactive client portal is embedded within the platform. All you have to do is to find the satellite images about a certain disaster, download them to your PC, select the path where you store the images, and upload both pre-disaster and post-disaster satellite images. Users are authorized to upload pre-disaster satellite imagery and corresponding post-disaster satellite imagery simultaneously and execute the algorithm to yield a result. Currently, the platform only supports JPG or PNG formats of images of the fixed size, and the result of the model will not be a preview of the web page but can be downloaded instead since a uniform input format and implicit output can significantly boost the process when the application programming interface is invoked to process the images in large batches.

4.3. Challenge 3: How to Solve the Concurrent Access Problem?

The high concurrent access always challenges the stability of the website. High single-serving time may make this problem more severe, especially for the heavy model of the remote-sensing application. Too many requests at the same moment cause GPU memory resource depletion and a bad user experience for waiting. There are technical details in the following to control the GPU usage and improve the user experience while waiting.

4.3.1. Control the GPU Usage: Release Resources and Maintain a Thread Queue

- Considering the limited GPU resources of our device, we have adopted some optimization upon the Pytorch framework. First, we minimize the unnecessary intermediate variables in our code. As an instance, using "a = 2a" instead of creating a new variable with "b = 2a" will save quite a lot of space. Moreover, releasing the image memory promptly and deleting the used image storage helps a lot in reducing the burden of GPU.
- To handle frequent and multiple requests, we maintain a task queue collecting tasks in chronological order. Instead of performing tasks serially, we turn on a multithreaded structure. Once a single request is started in a thread, the user will get a notification. Meanwhile, the web will frequently make inquiries about the server until the classified images are output.

4.3.2. Improve the Waiting Experience: Asynchronous Rendering and Polling by the JavaScript

The inference time of our current model is not fast enough and may not meet the needs of the industry application. It is not tolerant for any user to wait for a final result without any feedback. Moreover, the Hyper-Text Transfer Protocol (HTTP) limits the server sending data without requests from the browser. One way to improve the user experience for waiting is giving feedback by pooling each API and rendering any result as long as it has finished. Specifically, our JavaScript controls the browser request to the server every 0.1 s to discover whether the prediction result has been saved at the target path.

4.4. *Challenge 4: How to Design an AI Platform Easy for Data Scientists to Iterate the Algorithm?*

The major target for the platform is to present our latest research with a friendly interface rather than providing only the resource code and the checkpoint, which prevents the visitor from experiencing the latest model. For data scientists, Python is an easy and popular language to validate the idea and realize the algorithm. The best situation is to present our latest experiment model by simply updating the checkpoint and source code on the server.

4.4.1. Platform Structure Based on the Technology Stacks of the Python Family

To fully present our latest model and to provide a friendly interface to visitors, an online disaster building-damage-detection system is designed and developed, called AI-driven Damage Diagnosis Platform (ADDP). The main function of ADDP is targeted to directly present the building-damage-detection results and application potentiality of typical AI-driven models. The ADDP has a service-oriented architecture containing the following four layers: an application layer, a logit layer, a service layer, and a resource layer, as shown in the following Figure 12. The web structure and deep-learning structure are Django and Pytorch from the Python family, which makes it easy for data scientists to update the model from the back end.

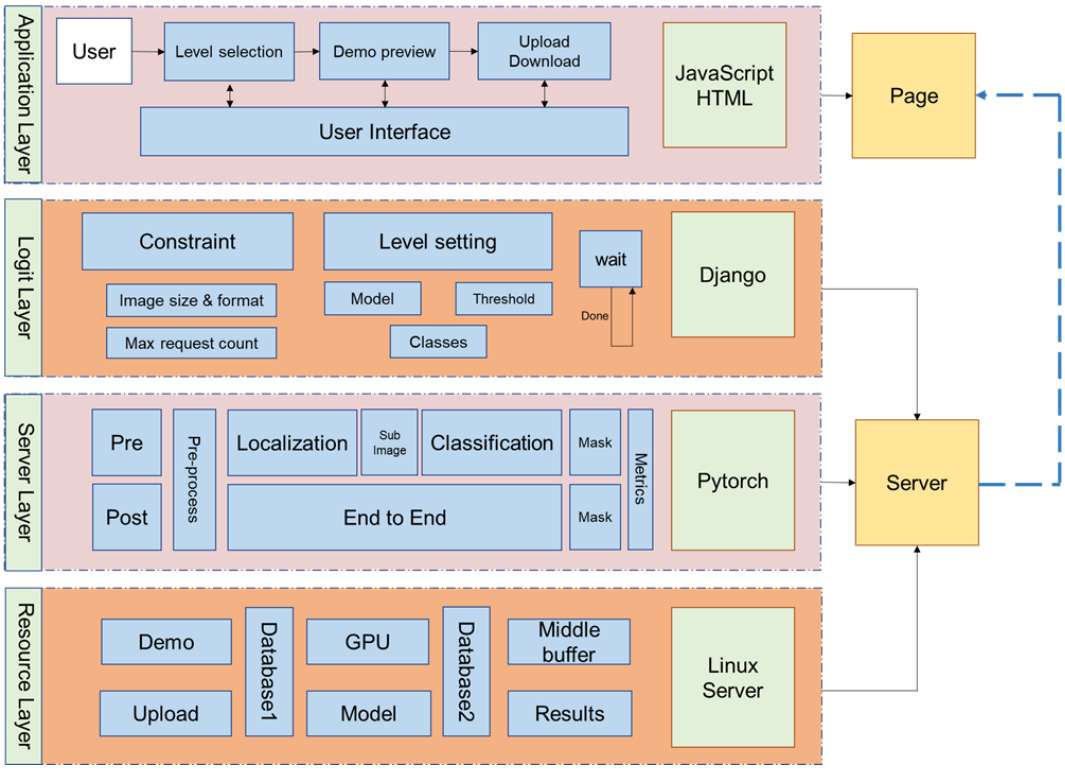


Figure 12. The overall structure of the AI-driven Damage Diagnosis Platform (ADDP). It has a service-oriented architecture containing the following four layers: an application layer, a logic layer, a service layer, and a resource layer.

The application layer lies at the top of the ADDP architecture and represents a friendly user interface that enables end users to select the model level and metrics threshold level, which helps to get a better building-level evaluation. For non-professional users, the platform designed a friendly demo preview interface, where users can select different demo images and select a rectangular area for prediction. As for the professional user, we provide upload and download API. This API gives users a chance to upload the latest remote-sensing image to test the model performance. This part is mainly developed with JavaScript. Since we apply a structure separating the front end and back end, JavaScript helps the user-end communicate with the model-end server.

4.4.2. Pipeline Design Specifically for Building-Damage Detection

The logit layer mainly helps users switch to a preferred mode and handle abnormal cases, hostile attacks, and concurrent assess. To get a valid sub-image from the demo image, the server will check the request parameter. Since the prediction of the model costs a few seconds, it is not good to run too many predictions and drain the computing resource in a short time. We build and maintain a task queue and make sure only 5 sub-threads run at the same time. This part is developed based on a Web Frame for Python. The detailed workflow of the logit layer is shown in Figure 13.

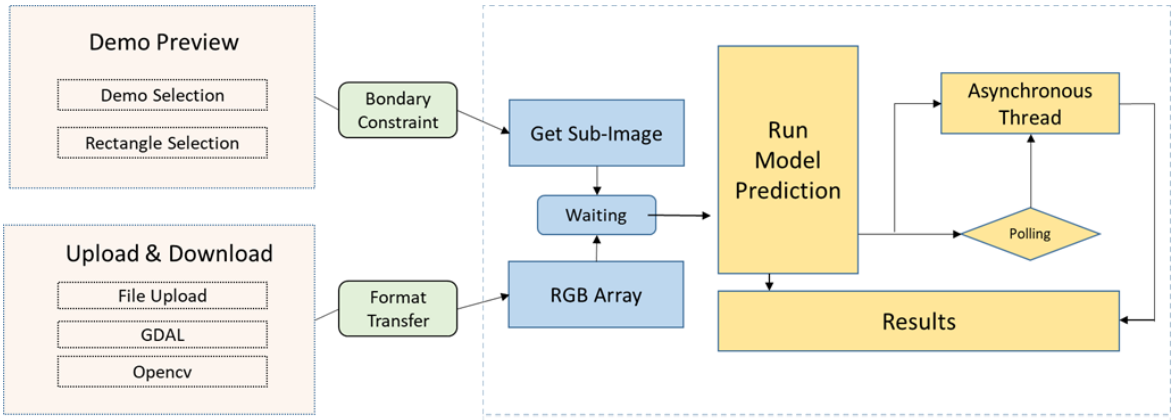


Figure 13. The detailed workflow of the logit layer we design. The structure helps users switch to a preferred mode and handle abnormal cases, hostile attacks, and concurrent assess by a task queue design which makes sure only 5 sub-threads run at the same time.

In the service layer, we use Pytorch to build up our AI-driver prediction pipeline. ADDP mainly provides AI semantic segmentation service. The building-level model indeed refers to two sub-modules. The localization module will give a building mask, and the mask can be split into several building polygons. The classification module can take building sub-images as inputs and give a damage type as the output. Moreover, the output can be the 5-class semantic segmentation result if the pixel-level end-to-end model is selected.

The resource layer lies at the bottom of the ADDP architecture and aims to provide a preview of computing devices, models, input datasets, and output database. Demo images and upload images consist of the input database, while middle results and final results are stored in the output database.

5. Conclusions

In short, this paper first reviews the work in the field of assessing building-damage and concludes mainstream methods into the building-level network and pixel-level network. Second, this paper introduces four key challenges and provides several solutions and corresponding case studies. Third, this paper introduces the platform which will continuously update the latest work and the state-of-the-art model for peers and visitors.

While reviewing the related work, the pixel-level model and building-level model are two mainstream methods for assessing building-damage tasks. Different methods and practical needs lead to four challenges, for which this paper gives corresponding solutions.

- Different metrics for the building-level and pixel-level put an obstacle in comparison. This paper puts forward the conversion method and a novel metric for comparison of the performance of different levels.
- The UAV is the most efficient device to get images after disasters, although it only captures the post-disaster image. This paper gives two solutions—one is naïve, and the other needs to use the GAN trained on the dataset with both pre- and post-disaster images.

- Disasters that can explicitly destroy a building happen infrequently, and severely destroyed buildings are relatively rare in the current open-source benchmark. This paper gives solutions from the perspective of both the data processing and the loss function.
- Real-time rescue demands faster inference of the damage situation with less computing resource in the industry environment. Feature-map subtraction, parameter sharing, knowledge distillation, and network pruning are discussed and studied by cases.

To comprehensively apply the above technical solution, an AI platform is developed for providing state-of-the-art results for assessing building-damage tasks.

6. Discussions

In the process of developing a web-based system for intelligent building-damage level assessment, we find that some problems remain largely unsolved. Further research in these fields is likely to promote intelligent building-damage-level assessment theories and applications.

There are plenty of brilliant approaches proposed to address the problem of damage-level assessment. Because of the diverse metrics used to evaluate the performance, the direct comparison of different models is largely invalid. A set of standard procedures for data processing and model evaluation is in demand so that the model with better performance on the task can be figured out. In this paper, we rethink some of the commonly used metrics and propose a method where F1 scores are evaluated by different IoU thresholds so that both the localization precision and the classification precision can be represented well in the final metric value. However, this method is not general enough to address various data structures and application situations.

Classification networks built on the xBD dataset often suffer from poor performance on "Minor Damage" and "Major Damage" class, primarily because of data imbalance in the dataset. Data sampling is an effective approach addressing data imbalance, which mainly falls into two categories, pixel level and image level. We introduce several specific methods in this paper, including both pixel-level ones and image-level ones, but we have not compared these approaches' efficacies. Further exploration can be made to figure out the relative efficiency of different data-sampling methods. Another problem concerning data imbalance in our task is that the Joint Damage Scale applied in the xBD dataset is subjective and sometimes indistinguishable on the boundaries of different classes. If the current four-class labeling is unnecessary in some cases, "Minor Damage" and "Major Damage" class could be converged to improve the comprehensive performance of the networks.

It should be noticed that the application of neural networks to intelligent building-damage assessment is quite different from developing novel model structures or data-processing methods in the laboratory. We need to pay attention to specific requirements concerning real-world situations. For instance, humanitarian rescue tasks emphasize rapid responses and value metric recall more than precision. There are also problems with the availability of data in practical applications. Concretely speaking, sometimes we must rely on drones for building images because there is no access to satellite images. Consequently, pre-disaster images may be unavailable, and the input image irregular boundaries may be misleading after the convolution operation. Although we discussed some solutions in this paper, many of these problems remain unsolved, and further research will be useful.

Author Contributions: Conceptualization, Y.B., J.S.; methodology, Y.B., J.S.; software, J.S., X.W.; validation, Y.B., J.S., X.W.; formal analysis, Y.B., J.S., D.L., X.W.; investigation, Y.B., E.M., S.K.; resources, Y.B., H.Y., E.M., S.K.; data curation, Y.B., E.M., and S.K.; writing—original draft preparation, Y.B., J.S., X.W., D.L.; visualization, J.S., X.W., D.L.; supervision, Y.B., B.Z., H.Y., E.M., S.K.; project administration, Y.B.; funding acquisition, Y.B., E.M., S.K. All authors have read and agreed to the published version of the manuscript.

Funding: This study was partly funded by the Fundamental Research Funds for the Central Universities, and the Research Funds of Renmin University of China (20XNF022), fund for building world-class universities (disciplines) of Renmin University of China, Major projects of the National Social Science Fund(16ZDA052), Japan Society for the Promotion of Science (JSPS) Kakenhi Program (17H06108), Core Research Cluster of Disaster Science and Tough Cyberphysical AI Research Center at Tohoku University.

Acknowledgments: This work was supported by Public Computing Cloud, Renmin University of China. We also thank the SmartData Club, an Entrepreneurship Incubation Team lead by Jinhua Su of Renmin University of China, Haoyu Liu, Xianwen He and Wenqi Wu, Students from Renmin University of China, Core Research Cluster of Disaster Science at Tohoku University (a Designated National University) for their support. We thank the two reviewers for their helpful and constructive comments on our work.The author gratefully acknowledges the support of K.C.Wong Education Foundation, Hong Kong.

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

Abbreviations

The following abbreviations are used in this manuscript:

IoU	Intersection over Union
MLOS	Main-Label Over-Sampling
DAC	Discrimination After Cropping
DAM	Dilation of11Area with Minority
SMOTE	Synthetic Minority Over-Sampling Technique
ADDP	AI-driven Damage Diagnose Platform
GCN	Graph Convolutional Network
ABCD	AIST Building Change Detection
FPN	Feature Pyramid Networks
R-CNN	Regions with CNN features
GAN	Generative Adversarial Network
CNN	Convolution Neural Network
RUS	Random Under-sampling

References

1. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA, 20–25 June 20 2009; pp. 248–255.
2. Koshimura, S.; Shuto, N. Response to the 2011 great East Japan earthquake and tsunami disaster. *Philos. Trans. Math. Phys. Eng. Sci.* **2015**, *373*, 20140373. [CrossRef]
3. Mas, E.; Bricker, J.; Kure, S.; Adriano, B.; Yi, C.; Suppasri, A.; Koshimura, S. Field survey report and satellite image interpretation of the 2013 Super Typhoon Haiyan in the Philippines. *Nat. Hazards Earth Syst. Sci.* **2015**, *15*, 805–816. [CrossRef]
4. Suppasri, A.; Koshimura, S.; Matsuoka, M.; Gokon, H.; Kamthonkiat, D. Remote Sensing: Application of remote sensing for tsunami disaster. *Remote Sens. Planet Earth* **2012**, 143–168. [CrossRef]
5. Gokon, H.; Koshimura, S. Mapping of building damage of the 2011 Tohoku earthquake tsunami in Miyagi Prefecture. *Coast. Eng. J.* **2012**, *54*, 1250006. [CrossRef]
6. Mori, N.; Takahashi, T. Nationwide post event survey and analysis of the 2011 Tohoku earthquake tsunami. *Coast. Eng. J.* **2012**, *54*, 1250001-1–1250001-27. [CrossRef]
7. Gupta, R.; Hosfelt, R.; Sajeev, S.; Patel, N.; Goodman, B.; Doshi, J.; Heim, E.; Choset, H.; Gaston, M. xbd: A dataset for assessing building damage from satellite imagery. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Long Beach, CA, USA, 16–20 June 2019.
8. Bromley, J.; Guyon, I.; LeCun, Y.; Säckinger, E.; Shah, R. Signature verification using a "siamese" time delay neural network. In Proceedings of the Advances in Neural Information Processing Systems 6, Denver, CO, USA, 29 November–2 December 1993; pp. 737–744.
9. Wheeler, B.J.; Karimi, H.A. Deep Learning-Enabled Semantic Inference of Individual Building Damage Magnitude from Satellite Images. *Algorithms* **2020**, *13*, 195. [CrossRef]
10. Trevino, R.; Sawal, V.; Yang, K. GIN & TONIC: Graph Infused Networks with Topological Neurons for Inference & Classification. 2020. Available online: http://cs230.stanford.edu/projects_winter_2020/reports/32621646.pdf (accessed on 20 November 2020).

11. Fujita, A.; Sakurada, K.; Imaizumi, T.; Ito, R.; Hikosaka, S.; Nakamura, R. Damage detection from aerial images via convolutional neural networks. In Proceedings of the Fifteenth IAPR International Conference on Machine Vision Applications (MVA), Nagoya, Japan, 8–12 May 2017; pp. 8–12.
12. Xu, J.Z.; Lu, W.; Li, Z.; Khaitan, P.; Zaytseva, V. Building damage detection in satellite imagery using convolutional neural networks. 2020. Available online: https://arxiv.org/pdf/1910.06444.pdf (accessed on 16 November 2020).
13. Huang, F.; Chen, L.; Yin, K.; Huang, J.; Gui, L. Object-oriented change detection and damage assessment using high-resolution remote sensing images, Tangjiao Landslide, Three Gorges Reservoir, China. *Environ. Earth Sci.* **2018**, *77*, 183. [CrossRef]
14. Nex, F.C.; Duarte, D.; Tonolo, F.G.; Kerle, N. Structural Building Damage Detection with Deep Learning: Assessment of a State-of-the-Art CNN in Operational Conditions. *Remote Sens.* **2019**, *11*, 2765. [CrossRef]
15. Mirza, M.; Osindero, S. Conditional generative adversarial nets. *arXiv* **2014**, arXiv:1411.1784.
16. Zhu, J.Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
17. Radford, A.; Metz, L.; Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv* **2015**, arXiv:1511.06434.
18. Isola, P.; Zhu, J.Y.; Zhou, T.; Efros, A.A. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
19. Caye Daudt, R.; Le Saux, B.; Boulch, A.; Gousseau, Y. Guided Anisotropic Diffusion and Iterative Learning for Weakly Supervised Change Detection. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019.
20. Nia, K.R.; Mori, G. Building Damage Assessment Using Deep Learning and Ground-Level Image Data. In Proceedings of the 2017 14th Conference on Computer and Robot Vision (CRV), Edmonton, AB, Canada, 16–19 May 2017; pp. 95–102.
21. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the 18th International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; Springer: Cham, Switzerland, 2015; pp. 234–241.
22. Humanitarian Data Exchange. Available online: https://data.humdata.org (accessed on 1 September 2019).
23. Cooner, A.J.; Shao, Y.; Campbell, J.B. Detection of Urban Damage Using Remote Sensing and Machine Learning Algorithms: Revisiting the 2010 Haiti Earthquake. *Remote Sens.* **2016**, *8*, 868. [CrossRef]
24. Ji, M.; Liu, L.; Buchroithner, M. Identifying Collapsed Buildings Using Post-Earthquake Satellite Imagery and Convolutional Neural Networks: A Case Study of the 2010 Haiti Earthquake. *Remote Sens.* **2018**, *10*, 1689. [CrossRef]
25. Hao, H.; Baireddy, S.; Bartusiak, E.R.; Konz, L.; LaTourette, K.; Gribbons, M.; Chan, M.; Comer, M.L.; Delp, E.J. An Attention-Based System for Damage Assessment Using Satellite Imagery. *arXiv* **2020**, arXiv:2004.06643.
26. Weber, E.; Kané, H. Building Disaster Damage Assessment in Satellite Imagery with Multi-Temporal Fusion. *arXiv* **2020**, arXiv:2004.05525.
27. Shao, L.; Zhu, F.; Li, X. Transfer Learning for Visual Categorization: A Survey. *IEEE Trans. Neural Netw. Learn. Syst.* **2015**, *26*, 1019–1034. [CrossRef]
28. Gupta, R.; Shah, M. RescueNet: Joint Building Segmentation and Damage Assessment from Satellite Imagery. *arXiv* **2020**, arXiv:2004.07312.
29. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 1–15.
30. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2014), Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
31. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 13–16 December 2015; pp. 1440–1448.

32. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, ON, Canada, 7–12 December 2015; pp. 91–99.
33. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 2672–2680.
34. Ali, A.; Shamsuddin, S.M.; Ralescu, A.L. Classification with class imbalance problem: A review. *Int. J. Adv. Soft Comput. Its Appl.* **2015**, *7*, 176–204.
35. Leevy, J.L.; Khoshgoftaar, T.M.; Bauder, R.A.; Seliya, N. A survey on addressing high-class imbalance in big data. *J. Big Data* **2018**, *5*, 42. [CrossRef]
36. Chawla, N.V.; Lazarevic, A.; Hall, L.O.; Bowyer, K.W. SMOTEBoost: Improving Prediction of the Minority Class in Boosting. In *Knowledge Discovery in Databases: PKDD 2003*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 107–119.
37. Cao, P.; Zhao, D.; Zaiane, O. An Optimized Cost-Sensitive SVM for Imbalanced Data Learning. In *Advances in Knowledge Discovery and Data Mining*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 280–292.
38. Hinton, G.E.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. *Comput. Sci.* **2015**, *14*, 38–39.
39. Liu, Y.; Cao, J.; Li, B.; Yuan, C.; Hu, W.; Li, Y.; Duan, Y. Knowledge distillation via instance relationship graph. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 7096–7104.
40. Zhao, B.; Tang, S.; Chen, D.; Bilen, H.; Zhao, R. Continual Representation Learning for Biometric Identification. *arXiv* **2020**, arXiv:2006.04455.
41. Li, Q.; Jin, S.; Yan, J. Mimicking Very Efficient Network for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
42. Liu, Y.; Chen, K.; Liu, C.; Qin, Z.; Luo, Z.; Wang, J. Structured knowledge distillation for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 2604–2613.
43. Geng, K.; Sun, X.; Yan, Z.; Diao, W.; Gao, X. Topological Space Knowledge Distillation for Compact Road Extraction in Optical Remote Sensing Images. *Remote Sens.* **2020**, *12*, 3157. [CrossRef]
44. Liu, Z.; Sun, M.; Zhou, T.; Huang, G.; Darrell, T. Rethinking the Value of Network Pruning. In Proceedings of the International Conference on Learning Representations (ICLR), Vancouver, BC, Canada, 30 April–3 May 2018.