Detecting Post Hurricane House Damage Using Geographic Information Related Multi-Resource Classification Model

Yihai Li[†]
Shanxi University
Institute of Mathematical Science & Applied Mathematics
Taiyuan, China
pokemonarrive@gmail.com

Shaotang Gu[*, †]
The University of Sydney (USYD)
Camperdown NSW 2006, Australia
School of Computer Science
[*]shgu2901@uni.sydney.edu.au
[†]These authors contributed equally.

Abstract. **Hurricane, like other natural cataclysms that threaten human life and houses' damage detection after a hurricane, is always a problem that needs to be solved. It is vital to retrieving the building damage status for planning rescue and reconstruction after the cataclysm. In this study, the convolutional neural networks (CNN) were utilized to identify collapsed buildings from post hurricane satellite imagery with the proposed workflow. Test accuracy (TeA), training accuracy (TrA), bootstrap algorithm, Grad-CAM, and feature maps (FM) were used as evaluation metrics. To overcome the imbalance, problems like overfitting, random flip, random sheer and zoom, and early stopping approach were tested on the investigations. The results demonstrated that the building collapsed information can be retrieved by utilizing post-event imagery. Simple convolutional neural network (SCNN) is the standard to compare the other two architectures, which achieved TrA 74.39% and TeA 76.91%, spend 18.22s per epoch. After adding an additional super resolution block specifically designed. The super resolution CNN with up sampling (SRCNN-US) reached lower TrA 78.01% but higher TeA 73.80% by spending nearly 4 times more (78.14s). Moreover, the multi input redesigned SCNN (MICNN) architecture showed better performance, with TrA value from 74.39% to 84.97% and TeA from 76.91% to 78.81% but consumed only 0.22s more per epoch. Combining MICNN and SRCNN-US, the MI-SRCNN-US model achieved the highest accuracy on the test set, 80.36%, and time-consuming, 83.50s/epoch. The 50 times bootstrap investigation shows that the MICNN makes predictions under more certainty with more accuracy. In subsequent evaluations, Grad-CAM and feature maps also prove that MICNN pays more attention to the building's region rather than its surroundings. Therefore, the suitable method to promote classification performance is by using post-hurricane cataclysm satellite imagery together with related geographic coordinates information as the input of CNN.**

Keywords: convolutional neural networks (CNNs); SCNN; image classification; geographic coordinate; multi- resource knowledge, multi resource CNN (MICNN), super resolution CNN with up sampling (SRCNN-US), Lota

I. INTRODUCTION

As the global climate continues to deteriorate, the frequency of some extreme weather events is gradually increasing. This severe natural disaster has caused incalculable damage to human lives and property. Conducting quantitative statistics on people and property affected by disasters has been a long-discussed issue. In the early days, people generally used manual methods to count the damage, which was too inefficient although more accurate. Although attempts have been made to increase the accuracy of the statistics by giving a quantitative analysis of the damage and developing some standardized testing like Forced Vibration Testing (FVT) [1], finding a method that is more efficient and better suited to the modern technological environment is what is being sought. With recent technological developments, satellites have acquired very high-resolution photographs, and then the invention and use of neural networks in the computer field. All these new technologies have given us new ideas and directions to solve problems. Recent advances in deep convolutional neural networks (CNNs) have led to remarkable progress in computer vision, especially in image classification. CNN involves many hyperparameters for identifying network structure, such as network depth, kernel size, number of feature maps, span, pool size, and pool area [2]. The combination of satellite images and convolutional neural networks is a new trend in post-disaster housing damage detection in recent years. Due to the superior performance of CNN in image recognition and the support of a large number of network frameworks (Alex-Net [3], VGG [4], ResNet50[5], DenseNet121[6], etc.), the use of off-the-shelf networks to do transfer learning on specific data and then to extract the edges of houses and determine whether they are damaged has been implemented with good performance [7]. Combining CNN with random forest can also effectively improve the accuracy of identifying affected buildings [8]. In this study, combining the CNN and additional data (Geo info) will be the main research direction.

High-resolution satellite images as a source of data for disaster-affected housing are currently a popular approach. Modern satellite images can contain a great deal of optical and other physical information, making the use of very high

resolution (VRH) images for image classification an important practical implement and trend. In recent years, convolutional neural networks (CNNs), one of the deep learning algorithms, have been widely used in the field of computer vision because of their superior performance. By adjusting and optimizing the hyperparameters (batch size, learning rate, and other additional factors), CNN can achieve better classification accuracy and less calculation time. Many enhancements have been made to the CNN to address the shortcomings of the current CNN [9] recently. For example, CNN struggles slightly with the problem of recognizing rotation-invariant images and adding a new rotation-invariant layer to the existing CNN architecture effectively optimizes the performance of CNN in processing rotational change images [10]. In dealing with the issue of analysis the condition of post-disaster houses, the method used is to create a dataset by comparing the VRH images acquired by satellite on both dates using a dual-date (pre-date and post-date of the affected area) and to extract the house damage features from the training dataset using the neural network.

From the current state of research, most projects on using a combination of neural networks and satellites image are still limited to the use of optical information from images of houses in VRH to create models for predicting the condition of houses. However, in addition to the highly accurate optical information obtained from the satellite images, there is also other meaningful physical information, such as the house's geographical location, which has some relevance to whether the house is affected or not. Often, in natural disasters such as hurricanes, they have a certain trajectory. The data mining method (HTPDM) was able to predict the hurricane's path with a predicted accuracy of 57.5% if it contains the match failure part, and the prediction accuracy would be 65% provided all matches are successful [11]. The geographical information (longitude, latitude) of the damaged houses gives a side view of the hurricane's path. Houses in a particular area of the map (where the hurricane has passed) are more likely to be damaged than houses in other areas of the map. Geographic information may be a relevant factor in predicting the damage of a house.

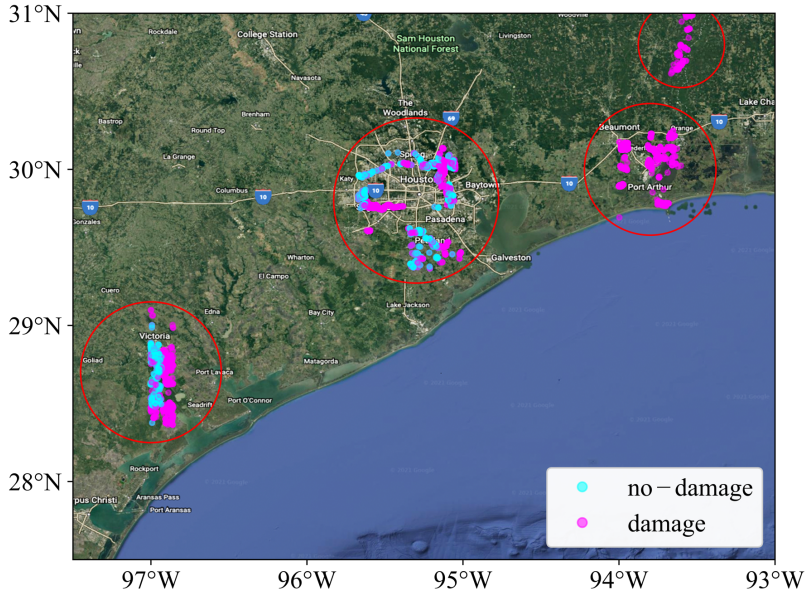

Fig. 1 The distribution of not damage/damage house display in the Houston area. The satellite map is generated from the google earth website [12]

Our research focuses on the use of both optical image data and geographic data from high-resolution satellite imagery in the network, and together participate in predicting whether a house is damaged or not. The network (The model with the best performance within three models) combined with geological data has 9 layers (3 convolutional layers, 3 max-pooling layers, 1 dropout layer, 1 dense layer, 1 dense location layer) and 1,063,173 parameters trained. The network achieved relatively good prediction results with a very simple structure compared to some networks like VGG deep neural network. That is one of the most advanced and powerful deep learning models commonly used in computer vision and quite costly to train and serve due to the nature of its large parameter set [13].

The remainder of this paper is structured as follows. Section 2 presents the basic principles on which our research is based. It is about the convolutional layer and machine learning. Section 3 describes what the data is used and what preprocessing works are done. The three kinds of network preview and brief methodological framework description are in Section 4. Finally, the result explanation and discussion are in Section 5. Section 6 is the conclusion part.

II. Problem Description

Based on existing satellite technology, a large number of satellite images can be produced every day. Processing information from these high-resolution images is a painstaking and time-consuming job without the help of computers and associated image processing algorithms. The demand constantly arises in the last decades of identity city components are destructed or stay safe. Therefore, the previous algorithm cannot extract essential information such as the edge of builds, circumstance around the city block, and the damage degree of certain built. The cause of that destruct such as a tornado,

hurricane, earthquake, or inferno became the obstacle that stopped people from making the next step. The use of neural network algorithms and the treatment of housing damage prediction as a classification problem has become a new breakthrough for this type of problem.

In this study, the image from the satellite is used to predict the damage of houses and train the networks designed in this article. The labeled satellite image recognition question could be demonstrated in an image classification problem. The images are separated into two categories: damage & not damage, and each of the images has been preprocessed into the same size-squared with 128 pixels on edge. For each input image, the models are designed to predict whether it belongs to the damage class or not damage class. The models aim to minimize the objective function, which is also called the loss function, the binary cross entropy loss[14].

$$H(p,q) = -\sum_{x}[\mathrm{p(x)ln(q(x)) + (1 - p(x))ln(1 - q(x))}] \quad (1)$$

where the H represents the total cross entropy of two probability distributions of the variable x, and the distribution function of x - $p(x)$, $q(x)$ represents the predicted probability that x is in the "damaged" class.

III. DATA GENERATION

These data are collected from "Geo-satellite sensor" and "Geo Bigdata", after the hurricane and before. The data of the Geo-satellite sensor is from GeoEye-1, which is capable of acquiring image data at 0.46 meter panchromatic (B&W) and 1.84 meters multispectral resolution. It also features a revisit time of fewer than three days and the ability to locate an object within just three meters of its physical location. The GeoEye-1 satellite sensor features the most sophisticated technology ever used in a commercial remote sensing system. This sensor is optimized for large projects, as it can produce over 350,000 square kilometers of pan-sharpened multispectral satellite imagery every day. GeoEye-1 has been flying at an altitude of about 681 kilometers and can produce imagery with a ground sampling distance of 46 centimeters, meaning it can detect objects of that diameter or greater.

TABLE I. DATASETS FORMATION

x		***y***	
Picture train	(10000, 128, 128, 3)	Label train	(10000,)
Location train	(10000, 2)		
Picture validation	(2000, 128, 128, 3)	Label validation	(2000,)
Location validation	(2000, 2)		
Picture test	(2000, 128, 128, 3)	Label test	(2000,)
Location test	(2000, 2)		
Picture test another	(9000, 128, 128, 3)	Label test another	(9000,)
Location test another	(9000, 2)		

Note: Dataset's formation: The data used in tasks are separated into different datasets. The dataset is divided into a training set, a validation set, and 2 test sets. The table shows the data size of each subset. Dataset train validation and test are balanced with equal images of each class which the number of each category is 50/50. Test another dataset is unbalanced. This dataset is aiming to evaluate the performance of prediction models. And the location data consists of the latitude and longitude of the center of the image in four-digit decimal.

During the late summer of 2013, the orbit altitude of the GeoEye-1 satellite sensor was raised to 770 Km/ 478 Miles. GeoEye-1 new nadir ground sample distance (GSD) is 46cm than the previous GSD of 41cm. [15, 16]. And the collected data will be utilized to do evaluation have summary information shows in Table I.

In total, there are 23,000 128×128 RGB satellite images with their geographical coordinate information in the dataset. The dataset is divided into a training set, a validation set, and 2 test sets. Dataset train validation and test are balanced with equal images of each class. Test another dataset is unbalanced with 8000 and 1000 images of damaged and undamaged classes, respectively. Location data consists of latitude and longitude. Figure 2 randomly selects some images from each damage and no-damage class, also their specific location.

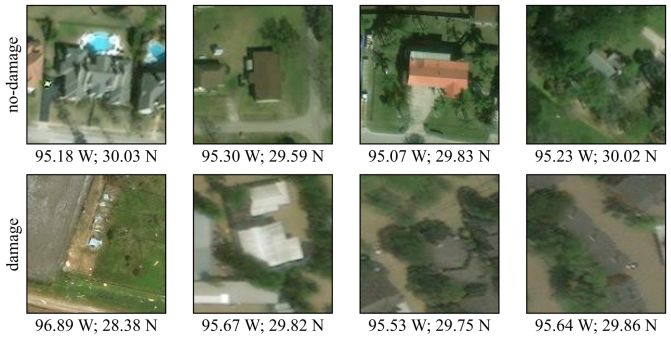

Fig 2. Originally collected data which already labeled with the damage/not damage, longitude, and latitude information. The figure shows 8 images total, including both damaged and undamaged houses in the dataset and the geographical information of the buildings in the images. It is easily observed that some of the damaged builds are surrounded by the water, and others are surrounded by forest after the cataclysm. For this level, constructions protected are always surrounded by more sophisticated circumstances.

In our work, some data preprocessing works are done for having better training. As normal preprocessing to avoid overfitting, normalization and shuffle the data randomly are also applied for improving the generalizability of certain training tasks. Moreover, the CNN model needs a large amount of data to guarantee learning efficiency and improve prediction accuracy. Overfitting problems also will often occur when the dataset is not large enough. Without previous processing, the data augmentation approach is a common method to inflate the training set. All in all, several data augmentation methods, including random rotation, width and height shift, shear range, zoom, horizontal flipping, are applied. Figure 3 randomly selected four different ultimately augmented images for each category. The changes in angle, size can be easily observed.

Fig 3. Images after augmentation include random rotation, width and height shift, shear range, zoom, horizontal flipping. The augmenting methods, which commonly generate undefined regions, use other approaches to fulfill the empty space. Sheer transformation is letting the undefined pixels directly equal to the edge of the image after the process. And these are what be seen in the images as straps.

Approaching this data augmentation by using the keras library function 'ImageDataGenerator' in python. Operate sheering and zooming method on each category's image with strength 20 radians. In the meanwhile, shift the image randomly on both horizontal and vertical with a range of 20 radians. In addition, we are randomly rotating the image with a range of 30 radians.

IV. Algorithms

Numerous existing methods are designed to achieve classification tasks in the long past path, like linear programming, *k*-means clustering, etc. Since the bloom of neural networks in the recent decade, especially in tasks similar to image classification.

To fix this problem, this article proposed three models to make the identification. The workflow can be seen in Table II. The beginning of one basic model's prediction is based on image input only comes with a simple structure of the classic convolutional neural network. Another is making predictions according to the satellite images and geological coordinates, similar to combining two neural networks. In addition, to evaluate whether the 128 square satellite pictures are sufficient enough or not, the third model has an additional block insert in the input and convolution layer, enhancing the image resolution two times. This method is achieved by adding a two-dimensional sampling layer block in the first basic model and then concatenating it with the original image. After that, all three models will be trained and evaluated separately on different datasets.

TABLE II. THE WORKFLOW OF IDENTITY COLLAPSED BUILDINGS

The Workflow
Collect Data
Imagery Patch
Building Damage Label Patch
Geographic Coordinates Patch
Augmentation
Normalization
Randomly Sheer,
Randomly Flip, Rotate, Zoom, Shift
Weight Balancing
Architecture Design
SCNN
MICNN, SRCNN-US, MI-SRCNN-US
Validation Metrics
TrA, TeA, Time Consumption
Bootstrap Experiment
Grad-CAM, Feature Map

A. *Simple Convolutional Neural Network (SCNN)*

The SCNN's architecture is designed to boost experiments processing faster and shorter and easier to modify, instead of pre-trained VGG or Res-Net, which is hard to evaluate the question this article aims to. Therefore, a simple network is suitable for use as a baseline model in this experimental process. The structure of the SCNN is displayed in Figure 4.

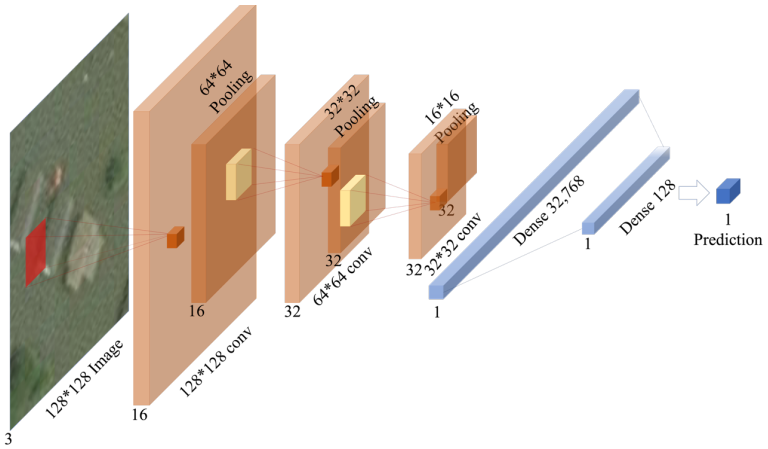

Fig 4. SCNN: The feature extraction layers are shown in Table III. The prediction layers are flattened layers and two dense layers, inserted by one drop out layer designed to prevent overfitting. Finally, end with one prediction layer output a probability.

Each layer in this network receives information from the previous layer of the network. The inputs are 128×128 RGB images, which go through only 3 convolution blocks, consisting of a convolution layer and a max-pooling layer. The pooling size is 2, which leads the image to half size. The pooling layer

operates upon feature maps and reduces input size to the following layers.

1) *2-D Convolutional Layer* The input of each convolutional layer is a $m \times n \times d$ feature map, where d is the number of the depth also channels, and $m \times n$ is the height and width of the feature map. The convolutional layers have K filters of size $s \times s \times d'$, where the size of s is smaller than min{m, n}, and $d' \leq d$, which is initialized at 3. The output is a $m' \times n' \times K$ feature map with depth K and size $m' \times n'$. Let W_k be the k-th filter and the x be the input feature map. The output k-th feature map, z_k, is computed using equation $z_k = f(W_k \bullet x + b_k)$. Where b_k is a trainable bias parameter and operator $\bullet$ is a 2-dimensional discrete convolutional multiply operator, and function f is nonlinearity function is applied after computing the output of the feature map.

$$f(x) = max(0, x) \quad (2)$$

For common use, this article applied the activation function as follows as formula (2). To maintain the size of the output is a certain size, usually dose zero-padding at the margin of the input feature map, which creates an edge of zero.

2) *2-D Pooling layer* The 2-D pooling layer simply takes the activations within quite small spatial regions of each feature map without repeat. Small spatial regions are commonly square, and a certain size is also called feature size. And then apply certain activation functions to extract features from spatial regions. In this article, the maximum activation function is also applied, and the feature size is as same as the 2-D convolution layer, but without the padding method.

3) *Prediction Layers* The prediction layers are composed of a couple of layers. With the feature extraction compound, the size of the feature map is not suitable for subsequent operations. To prevent the situation, a Flatten layer is required, which just flat a 3-D feature map, size $m \times n \times d$, to a single K-D vector v with a certain length, which $K = m \times n \times d$. And the Dense layer, also called the fully connected layer, takes the v as input, output as following formula $z_f = f(W_f \times v + b_f)$. W_f is the weight matrix size $K \times K'$, b_f is a trainable bias parameter, and f is the formula's activation function (2). The K' is the output size, bf has the size 1-K', and z_f is the output K'-D vector. The last dense layer is different of other dense layers with output size $K' = 1$, and the activation function is soft max activation formula [17] (3).

$$f(x) = \frac{1}{1+e^{-x}} \quad (3)$$

To prevent overfitting, with the data augmentation methods above are utilized, specifically, the drop out layer is applied before the last fully connected layer while the drop probability is 0.3 and behind the flattening layer inputted by previous pooling layer. Including all trainable parameters of network. The total number of model parameters is 1,063,169.

TABLE III. THE STRUCTURE OF THE SCNN

Layer Name	**Feature Extraction**			
	Filters num	**Filter size**	**Padding**	**Activation**
Conv2D	16	3	same	ReLU
Pooling2D		2	none	maximum
Conv2D	32	3	same	ReLU
Pooling2D		2	none	maximum
Conv2D	32	3	same	ReLU
Pooling2D		2	none	maximum
Layer Name	**Prediction layers**			
	Output Size	**Probability**		
Flatten	*32,768	-	-	-
Dense	**128	-	-	ReLU
Dropout	128	10%	-	none
Dense	1	-	-	Softmax

Note: *This data depends on the previous feature extraction layers, in this case, shown in the table: $32,768 = 32^3$. **This data is different in MICNN, which concatenates 2-D vector, makes 128 becomes 130. The structure of the first SCNN: has three convolutional layers and max-pooling layer with different filter numbers but the same filter size and padding approach. Convolutional layers have the same activation function called rectified linear function unit (ReLU). Finally, there are prediction layers that come after feature extraction layers.

B. Super-Resolution with Up-Sampling (SRCNN-US)

After shrinking down the input image into smaller size in CNN, there may be features blanked in the convolution layers process. To validate the hint, the second model is developed, by adding an additional bundle of layers between the input image and the first convolution layer, which contains a super resolution layer and convolution layer plus a max-pooling layer.

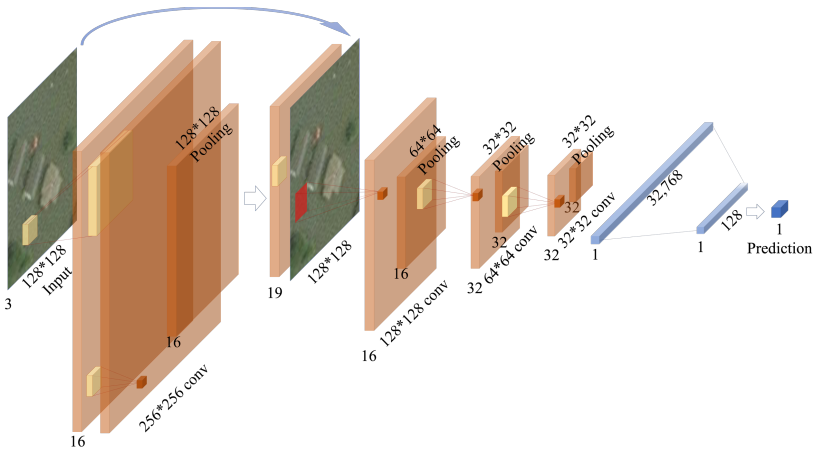

Fig 5. The SRCNN-US with the same prediction layers as the SCNN model shows in Figure 4. The up-sampling layer has a convolutional layer after and a max pooling layer, output shape is a 128 squared block with 1 depth, and them concatenates with the original image. The 128 squared blocks with 4 pixels depth outputted by super resolution block is put into the same feature extraction layer as SCNN.

In the super resolution layer, each 2 by 2 pixels block A in the image is transformed into a 3 by 3 block B. The process is marked as a matrix below

$$A=\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \rightarrow B=\begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \quad (4)$$

and for elements in matrix B the simplified formulas are shown below:

$$a_{11} = b_{11}, a_{12} = b_{13}, a_{21} = b_{31}, a_{22} = b_{33} \quad (5)$$

$$b_{i,2} = \frac{b_{i,1}+b_{i,3}}{2}, b_{2,i} = \frac{b_{1,i}+b_{3,i}}{2}, j \in \{1,2,3\} \quad (6)$$

To more edge occasion, the components of original matrix A rather spread them four elements to matrix B's each corner than place them at the corner of matrix B. The up-right corner case is showing in the demonstration formulas below.

$$B = B^t \quad (7)$$

$$a_{11} = b_{22}, a_{12} = b_{23}, a_{21} = b_{32}, a_{22} = b_{33} \quad (8)$$

$$b_{11} = 2b_{22} - b_{33} \quad (9)$$

$$b_{1,j} = 2b_{2,j} - b_{3,j}, j \in \{2,3\} \quad (10)$$

Applying the method on 3 channels and the output of the super resolution layer will be twice as large as the input image, which also has the same depth. The larger image will also process by a convolutional layer to extract necessary features, the number of filters is 16, and size of filters is 3, and the activate function is ReLU. Behind the convolutional layer also is a pooling layer with activate function shown in formula (2). The super resolution block outputs a 128 by 128 square with 16 depth which is the number of filters in the previous convolutional layer, and then concatenate with the original input, becoming a new 19 depth square matrix. Next, pass the matrix into the same feature extraction layers as SCNN. Finally, the prediction layers take the output as same as SCNN. Totally, the model has 1,065,921 trainable parameters.

C. Multi-Input CNN with geo-coordinate (MICNN)

Moreover, based on a previous idea rather than to use method to generate information, the third model is to use coordinates of each input image to optimize performance. The first is to concatenate the coordinates vector with flattened feature extraction from convolution layers.

This method has a magnificent upside in that it has nearly the same parameters as SCNN, which is only 8 additional parameters in the result. In MICNN with geo information, the imagery is processed in the same way as in SCNN. The location information is a 2-dimensional vector, and it enters a fully connected layer. This layer is concatenated with the last dense layer of the SCNN, generating a new fully connected layer containing both features extracted from satellite imagery and location information. To observe the influence of geological information come into the SRCNN-US model, the fourth model MI-SRCNN-US model combines both approaches from previous MICNN and SRCNN-US.

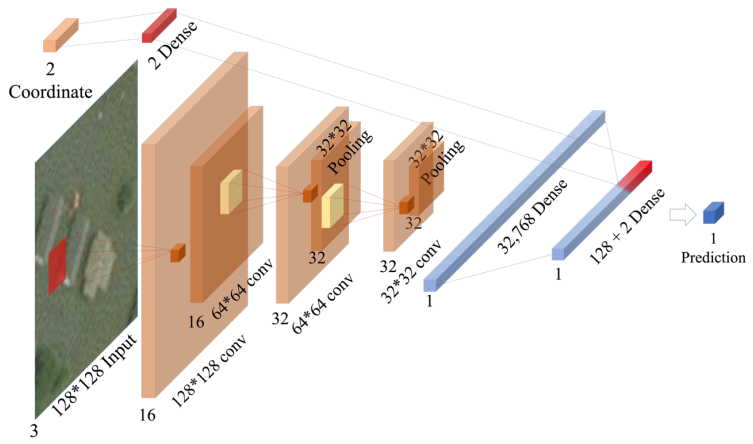

Fig 6. The MICNN: the third model has two inputs: the image matrix, which is similar to the other two, and the other is the geological coordinate vector containing latitude and longitude information. The additional vector will concatenate with flattened output from feature extraction layers after dropping out layer. The model will use the combined vector to make probability predictions.

For one perspective, the similarity of models is the loss function utilized is the cross-entropy loss function, and it is the same as the objective function in Section 2. Specifically, the $p(x)$ in prediction models is sustained in a constant number, 100%. And then, in neural networks' prediction, only $p(x) = 1$ is considerable, and then

$$H(p,q) = -\sum_x ln(q(x)) \tag{11}$$

so, the conditional probability of image x and class C is

$$P(C|x) = \frac{e^{Pc}}{\sum_j e^{Pj}} \tag{12}$$

Combine the equation (11) and the conditional probability formula (12), the total loss l (13) of the training batch is

$$l = \sum_i l_i\,(P,C) = -\sum_i \left[ln(\frac{e^{Pc}}{\sum_j e^{Pj}})\right] \tag{13}$$

P represents probability predicted by the network, and C is the actual class the data belong to. And another perspective, summation of trainable parameters of each model is also an item to justify. The total trainable parameters of each model are showed Figure 7.

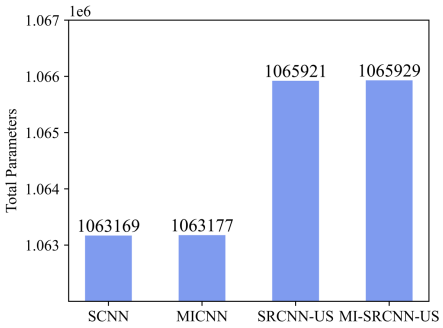

Fig 7. Total trainable parameters of four models.

MICNN and SCNN have approximately equal trainable parameters, which is nearly 1,063,100. But the SRCNN-US and MI-SRCNN-US have additional 2 thousand parameters in all, which may cause that the model needs more time to learn features.

D. Back Propagation Learning (BP)

The entire neural network can learn features by the BP of the loss function. Let $(\boldsymbol{x}_i, y_i)$ be the i-th training data, where $\boldsymbol{x}_i$ is the image data, y_i is in $\{0, 1\}$ is the true label of $\boldsymbol{x}_i$. $\boldsymbol{W}$ is the weights' parameters which be updated by minimizing the loss function, and the $l\,(\boldsymbol{x}_i, y_i, \boldsymbol{W})$ is the loss on the i-th training data. Compute the loss function gradient is the $J(\boldsymbol{W})$, throughout all training data, with formula (14)

$$J(\boldsymbol{W}) = \frac{1}{N}\sum_{i=1}^{N} l(\boldsymbol{x}_i, y_i, \boldsymbol{W}) + \lambda r(\boldsymbol{W}) \tag{14}$$

The loss function $J(\boldsymbol{W})$ throughout all of the training samples, where N is the number of the training sample, $r(\boldsymbol{W})$ is the regularization term, and λ is the decay coefficient, also called learning rate (LR). In the learning process, N can be sufficiently large ($n \ll N$). In practice, using a stochastic approximation of this objective in each BP iteration, and each epoch have $\frac{N}{n}$ mini-batches. To minimize the loss function, instead of using the Stochastic Gradient Decent (SGD) [18], the Adam[20] approach is utilized. Adam is an optimizer that combines the idea of root mean square back propagation and momentum. Instead of a single number λ represents the decay coefficient, Adam has a more complex updating algorithm.

Adam update algorithm begin with initialize parameters λ and $\rho_1, \rho_2 \in [0, 1)$. λ is the LR, and ρ_1, ρ_2 are the exponential decay rate. For current state t, firstly estimate first moment v_t and second moment r_t by formulas (15) and (16)

$$v_t = \rho_1 v_{t-1} + (1-\rho_1) J(\boldsymbol{W})_t \tag{15}$$

$$r_t = \rho_2 r_{t-1} + (1-\rho_2) J(\boldsymbol{W})_t^2 \tag{16}$$

And to perform bias-corrected first and second moment variables, use the following formulas (17) and (18)

$$v^{hat} = v_t / (1-\rho_1) \tag{17}$$

$$r^{hat} = r_t / (1-\rho_2) \tag{18}$$

Subsequent to above computations, update all weights and move to next state ultimately by formula (19)

$$W_t = W_{t-1} - \frac{v^{hat}}{\sigma + \sqrt{r^{hat}}} \lambda \tag{19}$$

The σ is a small number that making sure σ coefficient of λ does not become too large while training.

E. Evaluation Metrics

The method used to evaluate the performance of the three models is through the confusion matrix in Figure 8.

Fig 8. Confusion matrix (correct is green, incorrect is pink) The TP represents the number of Predicted Positives that were correct, and FP is the number of Predicted Positives that were incorrect. FN/TN is False Negative and True Negative, represent the number of Predicted Negative that were incorrect/correct. RP is the number of models that predict the result is Positive, RN is the number of models that predict the result is Negative. PP/PN is the number of actual results for Positive/Negative [19].

The predict accuracy is formula (20):

$$accuracy = \frac{TN+TP}{TN+FN+TP+FP} \tag{20}$$

To investigate all models in different perspectives, there are many methods used to do evaluations. Section A illustrates all training records and evaluations on the picture train dataset and picture test dataset. And then, Section B uses the bootstrap algorithm to evaluate models in a reasonable experiment. Finally, Section C demonstrates some maps inside models or generated by models like feature maps and Grad-CAM maps.

V. Result & Discussion

In this study, results are collected using the device with M1(2020) chip (8-core CPU with 4 performance cores and 4 efficiency cores, 8-core GPU, 16-core Neural Engine) and 16GB unified memory. The operating system is macOS Big Sur (Version: 11.2.2). The software platform is TensorFlow (Version: 2.4.0-rc0). And a sequence of CNN-based approaches was proposed for constructions collapsed assessment after catastrophe hurricane.

A. Identifying Damage Buildings Using CNNs

In the training process, algorithms and methods used to prevent overfitting are data augmentation and data balancing and include training strategies. In this essay, early stopping and reduce learning rate while accuracy or loss reaches the plateau. Specifically, monitoring the validating accuracy for both strategies. Early stopping's minimum delta of is 0.01, and patience is 5 epochs. Reducing the learning rate has a minimum delta of 0.00005 and patience 10 epochs and reducing it by divide 10. The history of training and validating accuracy of training process vs. epochs functions are showed in Figure 9.

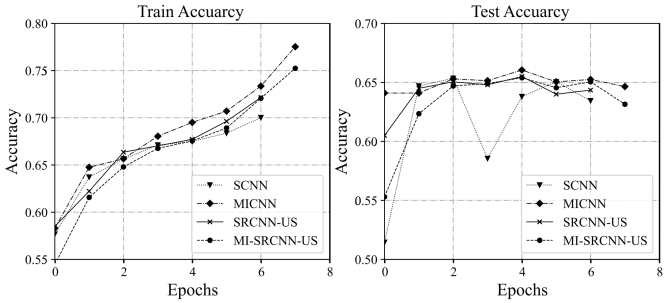

Fig 9. All three training processes were under parameters that batch size is 32 and started LR is 0.001, optimizers are both Adam with 10 epochs at least, but all training stopped before learning rate reduces. The training dataset is the picture train dataset and location dataset. The validation dataset is a picture validation dataset.

Figure 9 shows that these three models all have an overfitting phenomenon after about 4 epochs. In the end, models are performing well on the training dataset but unstable on the validation dataset. SCNN and MICNN stop process at 8-epoch but SRCNN-US stop earlier at 6-epoch. The other data showed in Table VI is the time consuming of each model in each epoch. The SRCNN-US took 78.14 seconds per training epoch, which is much less efficient than the other two models that took 18.22 seconds and 18.44 seconds, respectively.

TABLE IV. THE ACCURACY & TIME CONSUMPTION OF THREE MODELS.

	SCNN	MICNN	SRCNN-US	MI-SRCNN-US
TrA (%)	74.39	84.97	78.01	84.74
TeA (%)	76.91	78.81	73.80	80.36
Time (s)	18.22	18.44	78.14	83.50

Note: The accuracy of three models. Evaluation for all models is on picture test and location test dataset. And training evaluation is a picture dataset, time consuming for each epoch is an average number of total training time on epochs.

In a nutshell, Table IV demonstrates that the model with an up-sampling approach has a small improvement in TeA accuracy, which is about 0.95% compared to the SCNN. MICNN shows relatively high prediction accuracy on both TrA and TeA.

In the other dimension of Table IV, through the comparisons between the accuracy of the SCNN and MICNN, it can be observed easily that the accuracy of the model on the test set improved from 72.85% to 76.15% after adding the geographical information of the satellite images in the prediction process. This promotion also exists in the MI-SRCNN-US model and SRCNN-US model, the accuracy on the test set improved, from 70.80% to 80.36%. These results show that the additional geological components played a positive factor in predicting whether a building will be damaged or not, and also shows that hurricane damage to houses is regional in a natural location.

B. Performance of Bootstrap Experiments

To investigate them on this task further, a common way used a bootstrap algorithm to do analysis. This algorithm could prevent accidental influence that could make the model performance change noticeably. And it can, to some extent, guarantees that the model has as good a performance as possible. Saving the prediction results, observe distributions of certain

test individual data. Evaluate the performance based on the distributions. This approach is believable since the assumption that all neural model predictions have the same probability distribution that out use the certain self-learnable algorithm to approximate approach, and the Law of Large Number in classic Probability Theory [20].

Commonly, based on the algorithm, the number that chose to do bootstrapped training on an unbalanced test set is 50. This unbalance dataset (picture test another) could evaluate the model's performance in a real situation that the number of damaged buildings and no damaged buildings has a large gap. For this test, all models' weights have an additional weight to react in front of an individual category. The proportions are equal to the percentage of each category. The prediction of each time training is based on the picture test dataset.

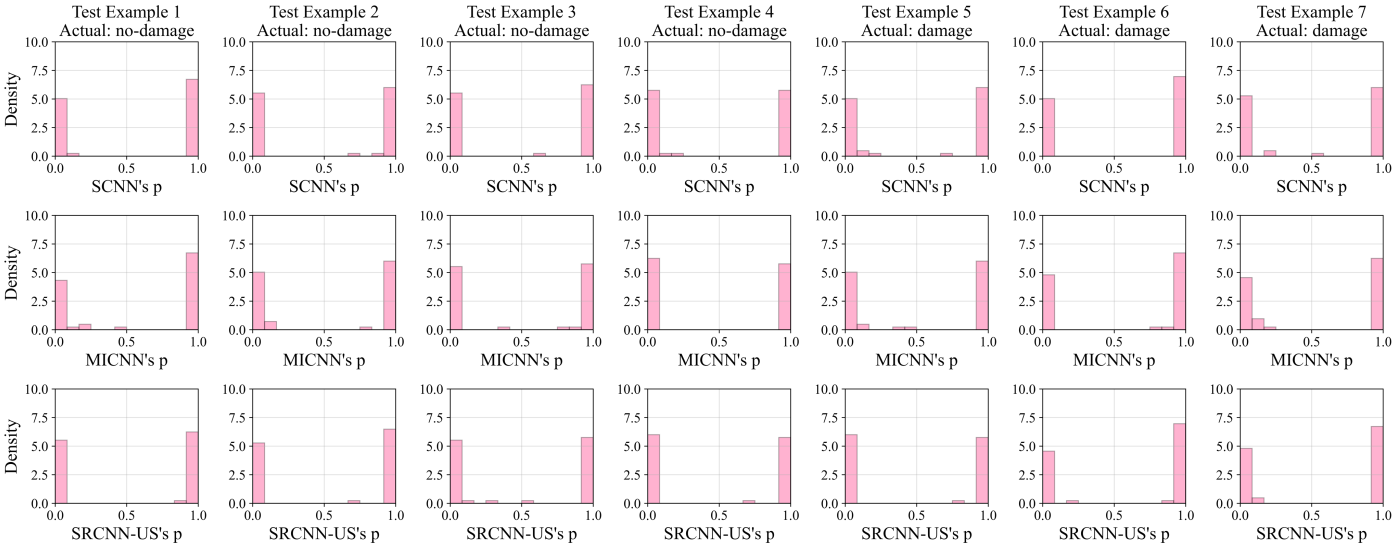

Fig 10. The sample from the 50 bootstrapped experiment results for three different networks. The *y* label called 'Density' demonstrates that the number of the *x* appeared in a specific region, and the *x* axis is the probability from predictions. The more columns stay at right or left, the higher accuracy that model predicts.

Next, randomly selected 7 individuals to demonstrate the performance of the 3 models. Plot the distribution for each observation, shows in Figure 10 and Figure 11. Regardless of the examples are used, all 3 models had a good performance with high degree of certainty. And then, it can be observed that test observations where the model predicts wrong also have great certainty. This may mean that model 2 had the best performance on the test set, may just an accident, and all 3 models' stability can be promoted.

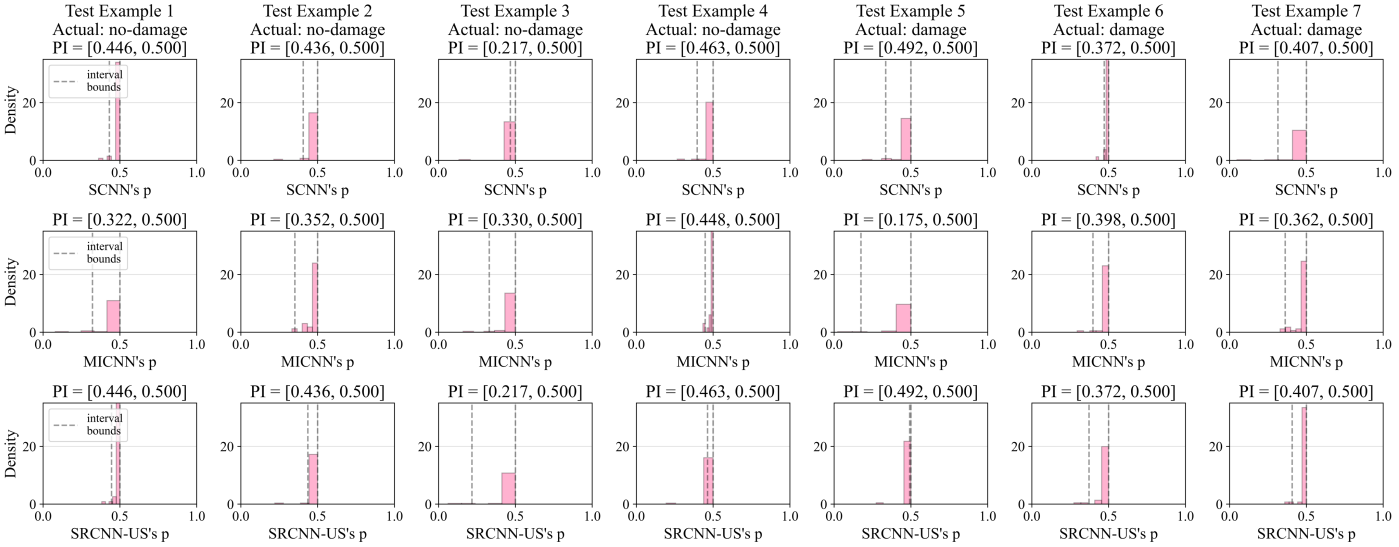

Fig 11. The sample from the 50 bootstrapped experiment results for three networks. The *y* label called 'density' demonstrates that the number of the *x* appeared in a specific region, and the *x* axis is the probability from predictions. This approach is trying to make columns in the center of the region [0, 1], and the bounds of prediction interval (PI) represents the certainty of model makes predictions.

All in all, take all 7 images displayed in Figure 10 and 11, SRCNN-US is more likely to make the wrong prediction with great certainty, but SCNN all 3 predictions are more likely to be right. The problem of SCNN is that it rarely can make the reverse decision, representing in the first row of Figure 10 and 11 that relatively large number of bars. The figure also

demonstrates that MICNN has the best performance, making all predictions more likely right with high certainty, among the 3 models.

C. *Inside CNNs Using Grad-CAM & Feature Maps*

Exploring the influences that are differences between three models brought, visualizing the Grad-CAM and features maps in the first convolutional layer, can help understand the correctness of the result. Take the randomly chosen illustrated image from datasets and compute the gradient flow back through each network and visualize the feature maps. The sample of feature maps in the meanwhile. Grad-CAM combined with the original image and the predictions generalize by three separated models are shown below in Figure 12.

Gradient-weighted Class Activation Mapping (Grad-CAM) uses the gradients of the target concept (building) flowing into the final convolutional layer to produce a coarse localization map highlighting the significant regions in the image for predicting the concept [21].

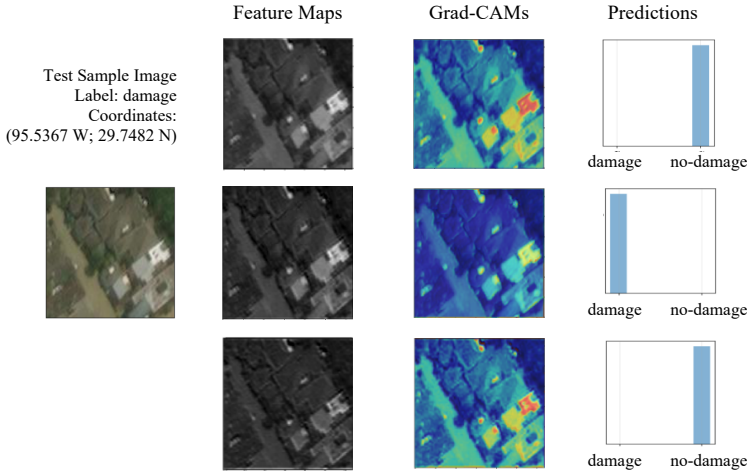

Fig 12. Grad-CAM & feature maps (Three models with the same image input). The sample image takes from the dataset randomly. The figure above shows that the gradient-weighted class activation mapping (Grad-CAM), feature maps from filters in the convolutional layer, and prediction of models and the up to down is SCNN, MICNN, and SRCNN-US. The grad-cam images are processed, which combined the original input image with the heat map generates by the Grad-CAM algorithm, shows the priority of the network pays attention.

From observation of the Grad-CAM in Figure 12, the part in the image that is important for predicting is quite obvious to observe relatively, through combined the heat-map and original image. Specifically, the left side is the actual image which is the input of three models. In the Grad-CAM row of Figure 12, the distribution of pixels of the second model appears more even and smooth and represents that model only pay attention to the building region rather than other useless information like surroundings and circumstance that often cause distraction.

By observing the original input images, the constructions that the hurricane hit are surrounded by water like an island in the ocean, damaged in their original geographical appearance. In the aftermath of the hurricane, the previous environment would have changed from ‘neat’ to ‘chaotic’, and the build would have been less distinguishable from its surroundings.

This illustrated that the MICNN redesigned based on the SCNN model is more accurate in predicting whether the house is damaged and is in line with our inference above. Compared to the other two neural models, as in the previous guessing, the model with up-sampling has the similar overall priority that pays on the image’s region, not as smooth as the MICNN model. This phenomenon is which could lead these models to incorrect prediction.

VI. Conclusions

Satellite imagery and CNNs are widely useful to investigate the post-event collapsed building. In this study, to compare the performance of different CNN structures, three redesigned CNN models are trained on the same datasets to identify collapsed buildings after a hurricane. Evaluating both of them on the picture test dataset, the accuracy of the SCNN is 76.91%. The MICNN with geological information enhanced the accuracy of the test dataset by at least 2.0% and only added 8 more parameters. SRCNN-US has a better performance than SCNN with nearly 1% accuracy. In all, the MI-SRCNN-US model has the highest accuracy, 80.36%, with the highest cost.

Furthermore, apart from accuracy, responding time consumption is also a vital issue in a cataclysm. Compared with object-oriented classification, the redesigned models are more efficient and more accurate. Since MICNN spends less than a quarter of SRCNN-US time to reach the best performance among the three candidate models. By comparing the testing accuracies and time consumptions among all models, it is easy to observe that designing a combination of the CNN model with additional geographical information does improve the model’s performance.

Meanwhile, without transfer learning, using pre-trained CNN models such as VGG, MICNN and SRCNN-US are also achieved a reasonable performance with a quite SCNN architecture. Considering performance, a combination of CNN and geographical information shows great potential in loss assessment after the disaster. In the long term, this idea could be applied to other advanced models to improve their performance by using the image to classify image category and include other types of information like coordinates.

REFERENCES

[1] Rosenblatt W, Raney J, Zavala A . Poly Canyon Bridge House- Damage Detection Using Forced Vibration Testing. 2015.

[2] Albelwi S, Mahmood A. Automated Optimal Architecture of Deep Convolutional Neural Networks for Image Recognition[C]// 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA). IEEE, 2016.

[3] Krizhevsky, Alex. One weird trick for parallelizing convolutional neural networks. arXiv preprint arX - iv:1404.5997 (2014)

[4] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. IJCV.

[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. CVPR, 2016

[6] Huang, Gao, et al. Densely connected convolutional networks. Proceedings of the IEEE conference on computer vision and pattern recognition. 2017

[7] Yang W, Zhang X, Luo P. Transferability of Convolutional Neural Network Models for Identifying Damaged Buildings Due to Earthquake. Remote Sensing. 2021; 13(3):504. https://doi.org/10.3390/rs13030504

[8] Ji M, Liu L, Du R, Buchroithner MF. A Comparative Study of Texture and Convolutional Neural Network Features for Detecting Collapsed Buildings After Earthquakes Using Pre- and Post-Event Satellite Imagery. Remote Sensing. 2019; 11(10):1202. https://doi.org/10.3390/rs11101202

[9] Soni, Ashish, et al. “Influence of Hyperparameter in Deep Convolution Neural Network Using High-Resolution Satellite Data.” Applications of Geomatics in Civil Engineering, Springer Singapore, 2019, pp. 489–500, doi:10.1007/978-981-13-7067-0_38.

[10] Gong Cheng, et al. “Learning Rotation-Invariant Convolutional Neural Networks for Object Detection in VHR Optical Remote Sensing Images.” IEEE Transactions on Geoscience and Remote Sensing, vol. 54, no. 12, IEEE, 2016, pp. 7405–15, doi:10.1109/TGRS.2016.2601622.

[11] Dong, X., and D. .. Pi. “Novel Method for Hurricane Trajectory Prediction Based on Data Mining.” Natural Hazards and Earth System Sciences, vol. 13, no. 12, Copernicus GmbH, 2013, pp. 3211–20, doi:10.5194/nhess-13-3211-2013.

[12] https://www.google.com.hk/maps

[13] Mai, A., Tran, L., Tran, L., & Trinh, N. (2020). VGG deep neural network compression via SVD and CUR decomposition techniques. 2020 7th NAFOSTED Conference on Information and Computer Science (NICS), 118–123. IEEE. https://doi.org/10.1109/NICS51282.2020.9335842

[14] Murphy K P. Machine Learning: A Probabilistic Perspective. MIT Press, 2012.

[15] Satellite Imaging Corporation.GeoEye-1 Satellite Sensor (0.46m), https://www.satimagingcorp.com/satellite-sensors/geoeye-1/

[16] MAXAR. GBDXTools. GBDX. https://www.geobigdata.io

[17] Han J, Moraga C . The Influence of the Sigmoid Function Parameters on the Speed of Backpropagation Learning. [C]// International Workshop on Artificial Neural Networks: from Natural to Artificial Neural Computation. Springer-Verlag, 1995.

[18] Bottou, L., 2010. Large-scale machine learning with stochastic gradient descent. In: Proceedings of International Conference on Computational Statistics. Physica-Verlag HD, Heidelberg, pp. 177–186.

[19] Powers, David M. W. (2011). "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation". Journal of Machine Learning Technologies. 2 (1): 37–63.

[20] Jaynes E T. Probability theory: The logic of science[M]. Cambridge university press, 2003.

[21] Selvaraju, Ramprasaath R., et al. “Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization.” International Journal of Computer Vision, vol. 128, no. 2, Springer Nature B.V, 2020, pp. 336–59, doi:10.1007/s11263-019-01228-7.