Post-Earthquake Serviceability Assessment Using the Convolutional Neural Networks and Deep Neural Networks

Author: Saiful Hasan, M.Tech (Earthquake Engineering), Jamia Millia Islamia , New Delhi, India.
Hasansaiful35@gmail.com
Co-Ordinator: Dr. SM Abbas, Senior Professor, Department of Civil Engineering, Jamia Millia Islamia, New Delhi, India. abbas_sm@hotmail.com

ABSTRACT

There have been studies using image processing techniques to assess damage of structure especially columns without sending individual to onsite. Convolutional neural networks (CNNs) and the entire Deep neural network (DNN) have shown state-of-art results in object detection and image classification tasks. This study proposed cascaded deep learning network for post-earthquake structural serviceability assessment. Major target deficiency components (crack, spalled area, transverse bar, and longitudinal bar) were used to determine the proposed damage states to assess serviceability of structure. Concrete surface cracks are major defect in civil structures. Structural Inspection which is done for the evaluation of rigidity and tensile strength of the building. Crack detection plays a major role in the building inspection, finding the cracks and determining the Structural health in the case of post-earthquake. The research paper is all about finding the serviceability results of the structure.

1. Introduction

Damages assessment of structures is important to derive an immediate response after severe events to decide serviceability. Especially, past earthquakes have proven the vulnerability of structure with insufficient detailing. Even modern reinforcement concrete (RC) structure/building and bridge columns, which are detailed properly to serve as the main source of ductility in a structure, may exhibit cover spalling, exposure of transverse and longitudinal bars, and buckling of longitudinal bars. To completely assess a structural column performance during an earthquake, both capacity of and demand on the columns are needed, which are usually in the form of displacements. (Kumar, 2014)

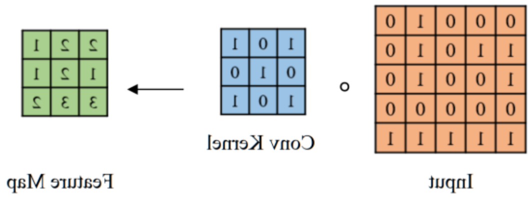

Figure 1.1: Basic concept of CNN

CNN is a feedforward network to extract features with convolutional formations. In figure 1.1 above, there is a general 3 × 3 convolutional kernel and 5 × 5 input image. CNN performs element-wise multiplication with input and convolution kernel and the results is called as a feature map. CNN kernels represent a different receptor that extract and derive useful features from the input source.

Figure 1.2 below is a sample of procedure of a two-dimensional CNN from. When we set the certain size of kernel, the border information can be lost. So, padding with certain value (in this example is 0) is applied to keep the border information and stride is applied to reduce calculation steps in convolving. After each convolution operation, it derives high dimension of features. But these feature maps can cause overfitting which has possibility of ending up where network only works with training data. So, pooling layer (down-sampling) is used to reduce overlapping information and in this example, max pooling which only keeps the maximum value in 2 × 2 window is introduced. And those hyperparameters (kernel size, max pooling window size, etc.) are designed and adjusted by each network configuration.

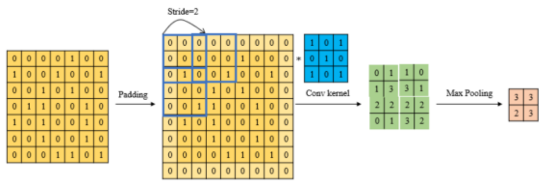

Figure 1.2: Procedure of a two-dimensional CNN

Due to lack of a national and unified post-earthquake inspection procedure for structure, conventional damage assessments are performed by sending professional personnel to the onsite, detecting visually and measuring the damage state. Although human-based assessment procedure may be effective, this procedure can take a lot of time and days after events and can miss critical time for rescue operations. And the correctness and accurate records of the decision for damaged structure may be different from subjectivity of the inspector. Also, several structural health monitoring (SHM) are capable to detect largescale damages in structure but used to be required with sensors or other instrumentals which are hard to install and not cost efficient. (BAZZURRO, 2017)

To get an accurate and fast damage results of a structure's condition is important to save not only lives but also costs. With the increasing demand of a computer vision-based method, automated damage detection has been developed more to help and make decision faster than past. However, as figure 1.2 shows, the task is non-trivial because in most sites, the damage appears with different shape and size, and usually mixed with noisy background which is hard to detect the parts which are needed to decide damage level.

In an early stage, there have been studies with heuristic filters to detect objects. Image processing methods with edge detection, threshold methods and traditional detectors were very popular in object detection.

Recently, deep-neural network (DNN) approaches (mainly convolutional neural networks (CNNs)) have shown to be promising for discovering the necessary features in works, such as collapsed-buildings identification. CNNs have also vastly been used in different applications, such as classifications, speech recognition, and the detection of various objects. Although plenty of studies have recently employed CNNs for damage detection, most of these techniques are designated for the detection of a maximum of two or three levels of classes, including undamaged, moderately damaged, and severely damaged. (Ogunjinmi, 2022)

Figure 1.3: Post-earthquake bridge column damage examples

Many studies investigated the performance of CNNs and different integrated techniques for recognizing damaged and undamaged areas from high-resolution remote imagery. Certain strategies—for example, a combination of CNN with features of point clouds—were carried out to identify the damaged areas through two classes of damaged and undamaged utilizing aerial photos. Another strategy of using DL in damage recognition could be employed by adding a field photograph to the aerial images to increase the accuracy of the damaged classes. A combination of multiscale segmentation with CNN is another approach to increase the classification accuracy of recognizing two classes of damaged and undamaged. Combining images with a different spatial resolution assisted by the capability of CNNs could also improve detecting damaged and undamaged classes. Utilizing pretrained CNN models (e.g., VGGnet), as well as data augmentation, proved to be an effective approach at detecting collapsed and no collapsed buildings. Moreover, texture features and CNNs for feature extraction and then utilizing different classifiers instead of CNNs could also improve the recognition of collapsed and uncollapsed buildings. The approach of combing CNNs with geographic information system (GIS) data (such as block vector data) could also play a positive role in recognizing three damage levels for groups of buildings block.

1. Literature review

Kim et al (2019) proposed the classification models using convolutional neural network (CNN) and speeded-up robust features (SURF) for crack detection. AlexNet and GoogleNet which both are state-of-art neural network models, have been applied to classify each crack and spalled image. Object detection methods have recently studied for a damage detection tasks. Object detection methods have improved an image classification tasks which classify entire images. The automatic approaches can identify the damaged building by extracting the defect features. However, different architectures, design methods, and widely changing real-world settings are still challenging these methods. Numerous studies have been done using standard ML approaches for building damage estimations post-earthquakes. (R Pani, 2019)

Yeum et al (2018) used Regions with CNN features (R-CNNs) for detecting and indicating objects with bounding boxes. Cha et al used Faster R-CNN which developed with a region-based method for detecting different shape and size of delamination. Besides those object detection methods, in our study, it is also important to quantify the damage to derive an accurate deficiency level. Semantic segmentation with object detection methods has been used to not only detect object with bounding rectangular box but also measure shape of the damage. Mask R-CNN has been adapted for detection of cracks, spalling and exposed bars. This method not only segments detected objects, but also provides the exact location of each instance in image. Using fully convolutional networks (FCN) also have been adapted to segment damages and based method to implement Mask R-CNN. It has been noticed the importance of detecting damage in structural level since this task can be a mixture of classification, object detection and semantic segmentation tasks. And deep learning-based methods have been studied to get state-of-art result for analyzing damages in column. (Anon., 2018)

Paal et al (2021) presented a computer vision-based method for determination of damage states of the column by localizing and quantifying each component (crack, spalling and exposed steel bar) properties of distinct textures of the region with Canny operator. Nishikawa et al. applied the multiple sequential image filtering for estimating property and detection. Yeum et al (2020) used region localization of object detection and filtering to detect fatigue cracks in steel bar. But those heuristic methods are time-consuming and cannot be operated in noisy background image. (Paal, 2021)

LeCun et al (2016) showed performance of CNN in classifying hand-writing digit dataset and the term “convolution” was first used. LeNet-5 is one of the earliest CNNs and the network was shallow model which only has 1 or 2 hidden layers. After advent of LeNet-5, many state-of-art networks has been inspired and created to solve complex classification and object detection tasks. Li et al stated many advantages of using CNN. First, there is a local connection which is different from previous multi-layer networks and these connections are very cost and time efficient by reducing the number of parameters. Second, a group of local connection shares the same weights, which accelerates to reduce calculation process. And lastly, pooling layer after feature map can reduce dimension of feature maps. Down-sampling in pooling layer can reduce less important data and remains only useful information. (LeChun, 2016)

2. Methodology

3.1 Dataset

The datasets contains images of various concrete surfaces with and without crack. The image data are divided into two as negative (without crack) and positive (with crack) in separate folder for image classification. Each class has 20000images with a total of 40000 images with 227 x 227 pixels with RGB channels. The dataset is generated from 458 high-resolution images (4032x3024 pixel) with the method proposed by Zhang et al (2016). High resolution images found out to have high variance in terms of surface finish and illumination condition. No data augmentation in terms of random rotation or flipping or tilting is applied.

3.2 Sample image

Figure 3.1: Sample image

In the general computer vision tasks, a large amount data and high-resolution images are required to drive state-of-art result. An insufficient number of data and low-quality images can lead poor performance and give hard time to train network model. In this study, approximately 40000 images were used for training and evaluation purpose of proposed network and divided into 80:20 ratios. Total training image number is 4000 and the number of evaluation image is 46. Each image has size is different, but the image resolution is at least 2000 × 1980 and contains each deficiency component (column, spalling, rebar, and crack). In addition, for generating extra data, data augmentation technique has been adapted to increase network training. Each image was augmented with left/right-side flip and Gaussian blur operation of standard deviation value 0.5.

3.3 Research Methodology

The general procedure for deep learning based structural condition assessment is: (1) collecting data; (2) labeling images; (3) defining the deep learning model; (4) training and testing the model, while some modification and customization steps are also included for structural condition assessment.

Mask TensorFLOW CNN and has been adapted to this proposed network to segment each target deficiency.

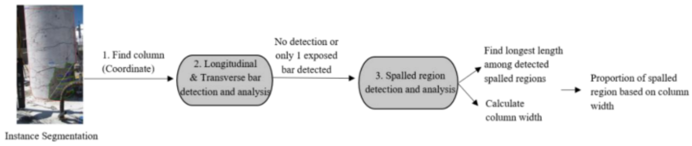

Figure 3.2: Target deficiency object analysis

Once the image is fed to the TensorFLOW CNN module, each targeted deficiency object in an image is segmented and masked. From this result, proposed network will follow those analysis steps to determine damage state. The results from this module contain each target deficiency's location, class, and mask. First, column instances are analyzed. Column instances are important since all damage states are evaluated by measuring and quantifying deficiencies within column area. From column's location (coordinate), it contains left-top x, y and right-bottom x, y and the area needed to inspect is narrowed by the box with that coordinate. Second, transverse and longitudinal bar analysis are followed by column detection phase. Proposed model counts the number of each detected bars and determine if counted number is matching with proposed damage state DS-5 or DS-4. If the model cannot find any components which match DS-5 and DS4, spalled region analysis step begins. In this step, the longest line inside of spalled region is used to determine damage state DS-4 or DS-3. For measuring longest line, mask information of spalled instance spalled region and measure the longest line inside of mask. For DS-3 and DS-4, column width is required to analyse those stages.

Since there is a column mask information from the results TensorFlow CNN module, RANSAC algorithm is used to calculate those two-vertical line of column. RANSAC algorithm is useful to fitting a line in two dimensions to a set of observations. Form masked column information, extract left-most-side coordinates and right-most-side coordinates. And RANSAC fit a line of each side and we can get a line equation for left and right side of column mask. And column width can be calculated by putting first y value of spalled region bounding box $[y1\ x1\ y2\ x2]$. And with this calculated width, the proportion between column width and longest distance is derived by simply dividing two values.

3. Results and discussion

As the training data sets in this case study include 40000 images, which is far less than the typical order of millions of images (e.g., in the ImageNet database), overfitting would be expected if the model was trained from scratch via such small data set. Considering these effects, the testing results were averaged on the 4,000 testing images over 100 epoch training episodes for each batch of training data and training strategies. The average testing accuracy results are given in table below. It is found from this study that as the size of the training data increases, the average testing accuracy keeps improving. It is recommended to have a training data size of at least several lakhs images if a good pre-strained CNN model is available for fine-tuning

4.1 Results

	Filepath	Label
0	../input/surface-crack-detection/Positive/0574...	POSITIVE
1	../input/surface-crack-detection/Positive/1870...	POSITIVE
2	../input/surface-crack-detection/Positive/0967...	POSITIVE
3	../input/surface-crack-detection/Negative/0791...	NEGATIVE
4	../input/surface-crack-detection/Positive/1400...	POSITIVE
...	...	...
39995	../input/surface-crack-detection/Positive/0854...	POSITIVE
39996	../input/surface-crack-detection/Negative/1944...	NEGATIVE
39997	../input/surface-crack-detection/Positive/0977...	POSITIVE
39998	../input/surface-crack-detection/Positive/1504...	POSITIVE
39999	../input/surface-crack-detection/Negative/1099...	NEGATIVE

40000 rows × 2 columns

Figure 4.1: surface crack detection input

Now, the code has defined the validation of images with the files names that has been defined below in the form of a table.

```
Found 3360 validated image filenames belonging to 2 classes.
Found 840 validated image filenames belonging to 2 classes.
Found 1800 validated image filenames belonging to 2 classes.
```

Figure 4.2: validation of images

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 02 | February - 2024 SJIF Rating: 8.176 ISSN: 2582-3930

```
Model: "model_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         [(None, 120, 120, 3)]     0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 118, 118, 16)      448
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 59, 59, 16)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 57, 57, 32)        4640
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 28, 28, 32)        0
_________________________________________________________________
global_average_pooling2d_1 ( (None, 32)                0
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 33
=================================================================
Total params: 5,121
Trainable params: 5,121
Non-trainable params: 0
_________________________________________________________________
None
```

Figure 4.3: Model table 1

```
Epoch 1/100
105/105 [==============================] - 20s 184ms/step - loss: 0.6845 - accuracy: 0.5639 - val_loss: 0.6278 - val_accuracy: 0.7071
Epoch 2/100
105/105 [==============================] - 19s 179ms/step - loss: 0.6066 - accuracy: 0.7136 - val_loss: 0.5113 - val_accuracy: 0.8202
Epoch 3/100
105/105 [==============================] - 19s 182ms/step - loss: 0.4946 - accuracy: 0.8243 - val_loss: 0.4160 - val_accuracy: 0.8833
Epoch 4/100
105/105 [==============================] - 19s 180ms/step - loss: 0.3909 - accuracy: 0.9000 - val_loss: 0.3275 - val_accuracy: 0.9179
Epoch 5/100
105/105 [==============================] - 19s 179ms/step - loss: 0.3061 - accuracy: 0.9172 - val_loss: 0.2647 - val_accuracy: 0.9440
Epoch 6/100
105/105 [==============================] - 19s 178ms/step - loss: 0.2456 - accuracy: 0.9438 - val_loss: 0.2152 - val_accuracy: 0.9476
Epoch 7/100
105/105 [==============================] - 19s 177ms/step - loss: 0.2004 - accuracy: 0.9502 - val_loss: 0.1942 - val_accuracy: 0.9607
Epoch 8/100
105/105 [==============================] - 19s 178ms/step - loss: 0.1599 - accuracy: 0.9564 - val_loss: 0.1881 - val_accuracy: 0.9405
Epoch 9/100
105/105 [==============================] - 18s 174ms/step - loss: 0.1457 - accuracy: 0.9591 - val_loss: 0.1662 - val_accuracy: 0.9643
Epoch 10/100
105/105 [==============================] - 19s 177ms/step - loss: 0.1410 - accuracy: 0.9645 - val_loss: 0.1819 - val_accuracy: 0.9321
Epoch 11/100
105/105 [==============================] - 19s 176ms/step - loss: 0.1347 - accuracy: 0.9610 - val_loss: 0.1385 - val_accuracy: 0.9750
Epoch 12/100
105/105 [==============================] - 19s 177ms/step - loss: 0.1139 - accuracy: 0.9640 - val_loss: 0.1425 - val_accuracy: 0.9714
Epoch 13/100
105/105 [==============================] - 19s 178ms/step - loss: 0.0935 - accuracy: 0.9730 - val_loss: 0.1246 - val_accuracy: 0.9750
Epoch 14/100
105/105 [==============================] - 18s 175ms/step - loss: 0.0901 - accuracy: 0.9743 - val_loss: 0.1385 - val_accuracy: 0.9774
Epoch 15/100
105/105 [==============================] - 19s 177ms/step - loss: 0.0920 - accuracy: 0.9769 - val_loss: 0.1160 - val_accuracy: 0.9774
Epoch 16/100
105/105 [==============================] - 18s 173ms/step - loss: 0.0994 - accuracy: 0.9700 - val_loss: 0.1217 - val_accuracy: 0.9762
Epoch 17/100
105/105 [==============================] - 19s 176ms/step - loss: 0.0927 - accuracy: 0.9732 - val_loss: 0.1167 - val_accuracy: 0.9774
Epoch 18/100
105/105 [==============================] - 18s 174ms/step - loss: 0.0825 - accuracy: 0.9754 - val_loss: 0.1272 - val_accuracy: 0.9786
```

Figure 4.4: Epoch iteration report

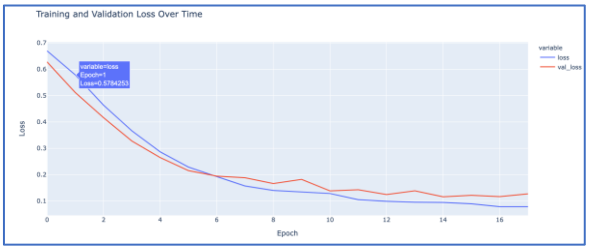

Figure 4.5: Training and validation loss over time at epoch 1

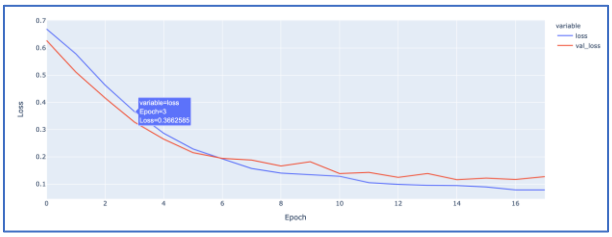

Figure 4.6: Training and validation loss over time at epoch 3

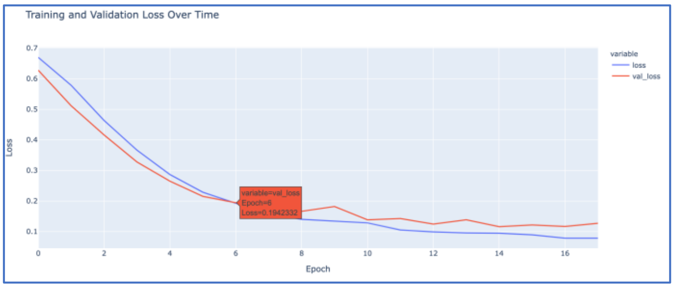

Figure 4.7: Training and validation loss over time at epoch 1

4.2 Comparison of validation loss and loss

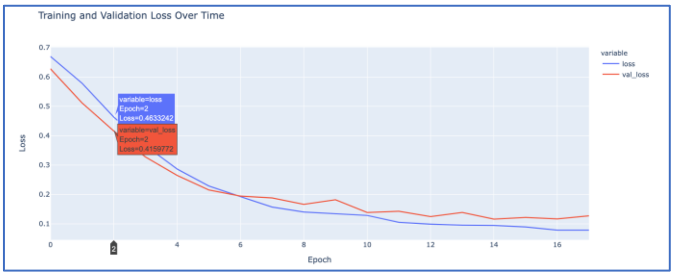

Figure 4.8: Training and validation loss over time comparison at epoch 2

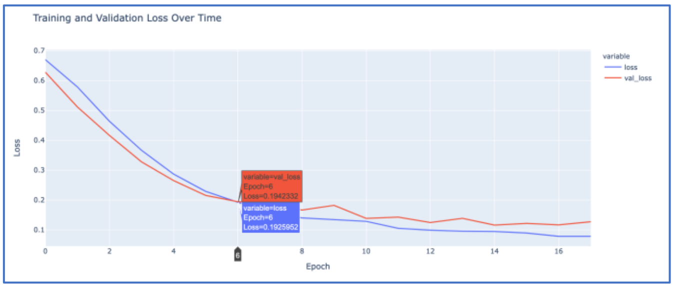

Figure 4.9: Training and validation loss over time comparison at epoch 6

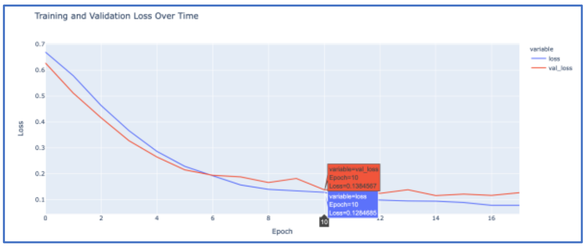

Figure 4.10: Training and validation loss over time comparison at epoch 10

4.3 Confusion matrix

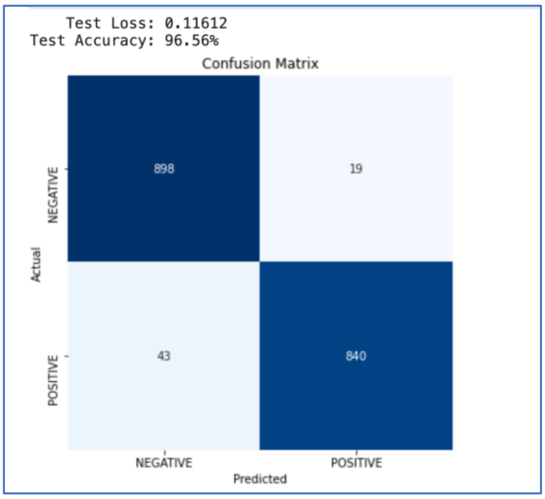

Figure 4.11: confusion matrix with test loss and test accuracy

5.Discussions

With the encouraging accuracy observed in the previous section, the potential of using the proposed method for brace damage detection appears to be promising. However, the reason for such success remains unclear. To address this question, the discussion in this section is attempting to interpret the results by visualization in order to understand why the CNN model can do well and how to optimize. The attempt was first trying to identify a recurring pattern through visual inspection of the results from the trained CNN model. This pattern was then used to formulate a handcrafted feature for classification using Bayesian decision theory, so the comparison can be made between the handcrafted feature and the feature identified by the trained CNN model in order to explain the reason for the high accuracy of brace damage detection. A modified version of the handcrafted feature and a feedforward neural network is also included for comparison.

The results from the trained deep learning model were visually assessed by using the guided TensorFLOW CNN method. The confusion matrix was plotted here at the end of the analyzation.

As discussed before, feature comparison results show that an ideal feature would be the one formulated to reduce the intersection area in the class-conditional probability density function curve plots. An example of this can be modifying the volume feature with a modifier (s), which was set as the scaling factor of the input earthquake ground motion with respect to the design earthquake level.

$$\hat{V} = s \cdot V = s \cdot \sum I = s \cdot \sum_{m=1}^{224} \sum_{n=1}^{224} I_{mn}$$

This modification is inspired by the fact that stronger earthquakes would have a higher likelihood to cause brace damaged in the concentrically braced frame (CBF) structure which is usually used in steel structure. Next, the activations in the last feature layer of the CNN model used in this study were extracted, an even better feature distribution can be qualitatively assessed. When the boundary was set as -0.38, the predicting accuracy per case was found to be 99.23% (healthy) and 98.41% (damaged), which were close to the estimated rate 99.19% (healthy) and 98.42% (damaged).
It was found that the trained deep CNN model achieved the best classification performance by generalizing the most discriminative feature distribution with the least intersection area, comparing to the features of volume, modified volume, and trained FNN. The difference between the trained deep CNN model and the FNN was found to be the definition of the feature function. In the FNN, the feature function was restricted to the form as expressed in Eqn above, while a more advanced feature function is believed to be generalized by the CNN model through stacking different computation layers. The advancement of the feature function had potential to increase the classification performance.

Conclusion

This study presented post-earthquake serviceability results using deep neural network and CNN. Mask R-CNN is composed of three stages (region proposal, classification, and segmentation). For instance segmentation of column, spalling and rebar, Mask R-CNN is trained with backbone network, and TensorFlow CNN has been used. Both networks have been trained with 4000 images of 100 epoch each and 1000 images for evaluation purpose.

In this dissertation I have taken the case study in the discussion section for the better understanding of the use of CNN in case of post-earthquake structural analysis. I took this as an opportunity to explore python to build a predictive supervised learning model. There were many aspects of the analysis which could have been extended such as a more extensive feature engineering, applying dimensionality reduction and hyperparameter turning. That said, without this an f1 score of 89.4% isn't that bad. Using the DNN technique I have found results which are helpful to get up to 97% idea damage in structure as well as I have got the confusion matrix also which is really interesting to understand the situation after the earthquake.

References

Kumar, R., 2014. *Damages assessment of structures is important to derive an immediate response after severe events to decide serviceability. Especially, past earthquakes have proven the vulnerability of structure with insufficient detailing. Even modern reinforcement conc,* s.l.: https://www.researchgate.net/publication/265338327_Effect_of_seismic_degradation_on_the_fragility_of_reinforced_concrete_bridges.

BAZZURRO, P., 2017. *Due to lack of a national and unified post-earthquake inspection procedure for structure, conventional damage assessments are performed by sending professional personnel to the onsite, detecting visually and measuring the damage state. Although human-base,* s.l.: https://www.iitk.ac.in/nicee/wcee/article/13_1708.pdf.

Ogunjinmi, P., 2022. *Recently, deep-neural network (DNN) approaches (mainly convolutional neural networks (CNNs)) have shown to be promising for discovering the necessary features in works, such as collapsed-buildings identification. CNNs have also vastly been used in differe,* s.l.: https://www.mdpi.com/1424-8220/22/9/3471/pdf?version=1651570091.

R Pani, K. e. a., 2019. *Kim et al (2019)proposed the classification models using convolutional neural network (CNN) and speeded-up robust features (SURF) for crack detection. AlexNet and GoogleNet which both are state-of-art neural network models, have been applied to classify e,* s.l.: https://link.springer.com/article/10.1007/s11831-021-09639-x.

Anon., 2018. *Yeum et al (2018) used Regions with CNN features (R-CNNs) for detecting and indicating objects with bounding boxes. Cha et al used Faster R-CNN which developed with a region-based method for detecting different shape and size of delamination. Besides thos,* s.l.: Yeum.

Paal, 2021. *Paal et al (2021) presented a computer vision-based method for determination of damage states of the column by localizing and quantifying each component (crack, spalling and exposed steel bar) properties of distinct textures of the region with Canny opera,* s.l.: https://journals.sagepub.com/doi/abs/10.1177/1475921719892345.

LeChun, 2016. *LeCun et al (2016) showed performance of CNN in classifying hand-writing digit dataset and the term "convolution" was first used. LeNet-5 is one of the earliest CNNs and the network was shallow model which only has 1 or 2 hidden layers. After advent of Le,* s.l.: https://www.ijirset.com/upload/2018/june/56_Survey_IEEE.pdf.