Eleventh U.S. National Conference on Earthquake Engineering
Integrating Science, Engineering & Policy
June 25-29, 2018
Los Angeles, California

DEEP LEARNING FOR AUTOMATED IMAGE CLASSIFICATION OF SEISMIC DAMAGE TO BUILT INFRASTRUCTURE

B. Patterson[1] , G. Leone[1], M. Pantoja[1], and A. Behrouzi[2]

ABSTRACT

The amount of structural damage image data produced in the aftermath of an earthquake can be staggering. It is challenging for a few human volunteers to efficiently filter and tag these images with meaningful damage information. The proposed solution is to automate post-earthquake reconnaissance image tagging activities by training a computer algorithm to classify each occurrence of damage per building material and structural member type. The approach is based on deep learning (DL), a subset of machine learning loosely based on the operation of a biologic neural system, which aims to learn and extract accurate representations from large data sets. DL algorithms are data driven; improving with increased training data. Thanks to the vast amount of data available and advances in computer architectures, DL has become one of the most popular image classification algorithms producing results comparable to and in some cases superior to human experts. The authors implemented a DL algorithm to automatically identify multiple damage types and associated structural members in a single image by adapting a pre-trained deep residual network. The algorithm was tested as follows: (i) binning building images as damage-no damage (88% accuracy), (ii) drawing a bounding box around damage in buildings (85% accuracy) and short/captive reinforced concrete columns with shear damage (77% accuracy). The lower accuracy of correctly identifying a target region in an image (test ii) compared to simple binning (test i) is anticipated since it is a more complex problem and there is a more limited number of expertly tagged training images (200 count) for shear damage-short column condition being studied. The research team expects algorithm accuracy will improve with training on additional images tagged for certain damage-structure pairs by a diverse set of experts.

[1]Dept. of Computer Science & Software Engineering, California Polytechnic State University – San Luis Obispo, San Luis Obispo, CA 93407 (email: mpanto01@calpoly.edu)
[2] Dept. of Architectural Engineering, California Polytechnic State University – San Luis Obispo, San Luis Obispo, CA 93407 (email: behrouzi@calpoly.edu)

Patterson B, Leone G, Pantoja M, Behrouzi A. "Deep Learning for Automated Image Classification of Seismic Damage to Built Infrastructure." *Proceedings of the 11th National Conference in Earthquake Engineering*, Earthquake Engineering Research Institute, Los Angeles, CA. 2018

Eleventh U.S. National Conference on Earthquake Engineering
Integrating Science, Engineering & Policy
June 25-29, 2018
Los Angeles, California

Deep Learning for Automated Image Classification of Seismic Damage to Built Infrastructure

B. Patterson[12], G. Leone[1], M. Pantoja[1], and A. Behrouzi[2]

ABSTRACT

The amount of structural damage image data produced in the aftermath of an earthquake can be staggering. It is challenging for a few human volunteers to efficiently filter and tag these images with meaningful damage information. The proposed solution is to automate post-earthquake reconnaissance image tagging activities by training a computer algorithm to classify each occurrence of damage per building material and structural member type. The approach is based on deep learning (DL), a subset of machine learning loosely based on the operation of a biologic neural system, which aims to learn and extract accurate representations from large data sets. DL algorithms are data driven; improving with increased training data. Thanks to the vast amount of data available and advances in computer architectures, DL has become one of the most popular image classification algorithms producing results comparable to and in some cases superior to human experts. The authors implemented a DL algorithm to automatically identify multiple damage types and associated structural members in a single image by adapting a pre-trained deep residual network. The algorithm was tested as follows: (i) binning building images as damage-no damage (88% accuracy), (ii) drawing a bounding box around damage in buildings (85% accuracy) and short/captive reinforced concrete columns with shear damage (77% accuracy). The lower accuracy of correctly identifying a target region in an image (test ii) compared to simple binning (test i) is anticipated since it is a more complex problem and there is a more limited number of expertly tagged training images (200 count) for shear damage-short column condition being studied. The research team expects algorithm accuracy will improve as with training on additional images tagged for certain damage-structure pairs by a diverse set of experts.

Introduction

Advances in technology, particularly smart phones with network connectivity, have facilitated widespread image data collection and immediate dissemination following recent seismic events. As a direct response, the Earthquake Engineering Research Institute (EERI) has transitioned to a new model of reconnaissance where the first line of response involves activating virtual response

[1]Dept. of Computer Science & Software Engineering, California Polytechnic State University – San Luis Obispo, San Luis Obispo, CA 93407 (email: mpanto01@calpoly.edu)
[2] Dept. of Architectural Engineering, California Polytechnic State University – San Luis Obispo, San Luis Obispo, CA 93407 (email: behrouzi@calpoly.edu)

Patterson B, Leone G, Pantoja M, Behrouzi A. "Deep Learning for Automated Image Classification of Seismic Damage to Built Infrastructure." *Proceedings of the 11th National Conference in Earthquake Engineering*, Earthquake Engineering Research Institute, Los Angeles, CA. 2018

teams that begin online data mining via formal and social media platforms for text, image, and video information. These volunteers compile data to inform the activities of deploying reconnaissance teams as well as the greater earthquake engineering community. The amount of data generated by a single major seismic event can be staggering and the responsibility for a few human data-gatherers to locate, identify, organize, and summarize the damage information in a meaningful and efficient manner is challenging. This paper presents a solution to automate virtual post-earthquake reconnaissance activities by training a computer algorithm to classify images for each specific occurrence of damage per building material and structural member type. The approach is based on deep learning (DL) [1], a subset of machine learning for image classification problems that makes use of a library of training examples to develop a robust and automatic visual recognition classification algorithm.

Per building material category (concrete, steel, masonry, timber, etc.), the assessment of post-earthquake damage requires the correct manual classification of around 20 structural member types and potentially 10 associated damage types. The authors have determined that, at a minimum, training of the DL algorithm requires about 200 images per damage-structural member pairing. Therefore, at least 40,000 images must be correctly tagged and subsequently verified by a team of experts. Gathering 200 images containing a clear illustration of an individual damage-structural member pairs has proven to be a much slower task than originally anticipated. Therefore, the research team has concentrated on proof of concept for DL by focusing on three different built infrastructure classification problems which have already shown promising results: (i) binary binning of building images as damage-no damage, (ii) drawing a bounding box around damage in buildings (as an extension of binary binning in task (i)), and (iii) drawing a bounding box around the specific damage-structure pair of short/captive reinforced concrete columns with shear damage.

The authors also implemented a graphical user interface (GUI) wrapper for the DL solution such that any engineer or researcher can train and deploy a neural network to classify specific damage-structural member pairs pertinent to their data. As a result, the GUI wrapper enables non-programmers to use their tagged image sets to contribute to the advancement of machine learning for computer vision in civil infrastructure. The most prevalent example where community-based efforts have led to the development of a large annotated image database is ImageNet [2] with over ten million tagged images and at least one million images where bounding boxes are also provided. The field of object classification has benefited tremendously from the pioneering scientists that released ImageNet and inspired others' contributions. A long-term objective of the research team is to collaborate with others to create a similar visual database for civil infrastructure, and specifically, damage types observed in civil infrastructure.

Previous Work

Brilakis *et al.* [3] implemented a visual pattern recognition (VPR) framework to identify and analyze visual features in images of structural members by translating shape and texture into numerical representations. The authors report good accuracy in distinguishing cracked versus uncracked concrete columns and identifying the location of exposed reinforcement in concrete members that have experienced cover spalling. The shortcoming with this type of classical computer vision approach is a lack of robustness, such that for more complicated damage patterns like drawing bounding box around concrete spalling regions there is a lower success rate.

Feng *et al.* [4] attempts to address the challenge of having an adequately large, expertly tagged image training set. The research team developed a deep residual neural network (based on the ResNet architecture from He *et al.* [5]) to maximize performance of a detection algorithm for civil infrastructure that targets four defects in an input image patch: cracking, deposit, water leakage, or any combination of the previous defects. The algorithm was trained using a small set of images (~600) annotated at the pixel level, and retraining the algorithm with newly available expertly tagged images along with images automatically classified by the algorithm. This process, while exhibiting relatively high accuracy for the four defects, is susceptible to feedback errors.

Yeum [6] developed a convolutional neural network algorithm (based the AlexNet architecture from Krizhevsky *et al.* [7]) to recognize post-hazard structural damage in reconnaissance images. The damage classifications were collapse-no collapse (binary) and concrete spalling/flaking (bounding box). A major aim of this project is to help engineers to filter collected images to facilitate analysis of specific structural damage types and in reconnaissance report writing.

Koch [8] presents an overview of the state-of-the-art (as of 2014) in computer vision-based defect detection in civil infrastructure, primarily: reinforced concrete bridges, tunnels, underground piping systems as well as asphalt pavements. There are only brief mentions of DL. The existing literature for computer vision in civil engineering concentrates on a small subset of defect types and utilize feature extraction approaches that makes the algorithms very specific and difficult to scale. In recent years, a new wave of computer vision algorithm based in DL makes automatic classification an easier task by embedding the feature identification within a learning pipeline.

Implementation of Deep Learning Algorithm

Deep learning algorithms are loosely based on biological neural systems where an individual neuron executes a very simple operation and sends the output signal to the rest of the neurons. One neuron does very little, but as a network they can perform extremely complex tasks. In computer science, neurons are simulated as simple software functions connected in groups (layers) by simple passing input/output arguments with varying weights assigned to each function. An interconnected layer performs a simple feature extraction to identify one higher-level feature in the image and by connecting many layers it is possible to identify entire objects in an image. Deep learning is a neural network (NN) approach with *numerous* different interconnecting layers. With such frameworks, DL can model very complex input (given many hundreds to thousands of inputs) to a specific output. This allows researchers to shift from problem dependent feature extraction to a more general DL algorithm.

Development of NNs involves two stages: (i) training on a set "ground truth" images that contain data known to be correctly classified and determining the optimal weight for the neurons, and (ii) deploying the NN using the learned weights from the training stage to classify a new image. Fig. 1 illustrates the steps in the training and deployment stages of a neural network workflow.

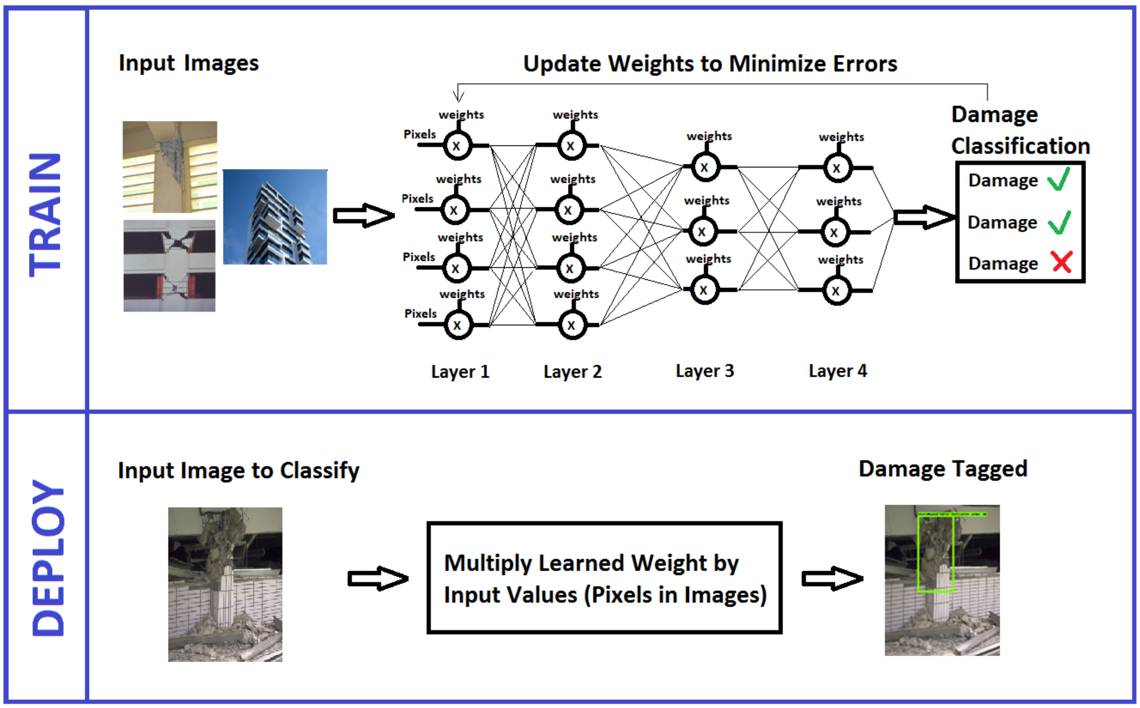

Figure 1. Deep learning workflow.

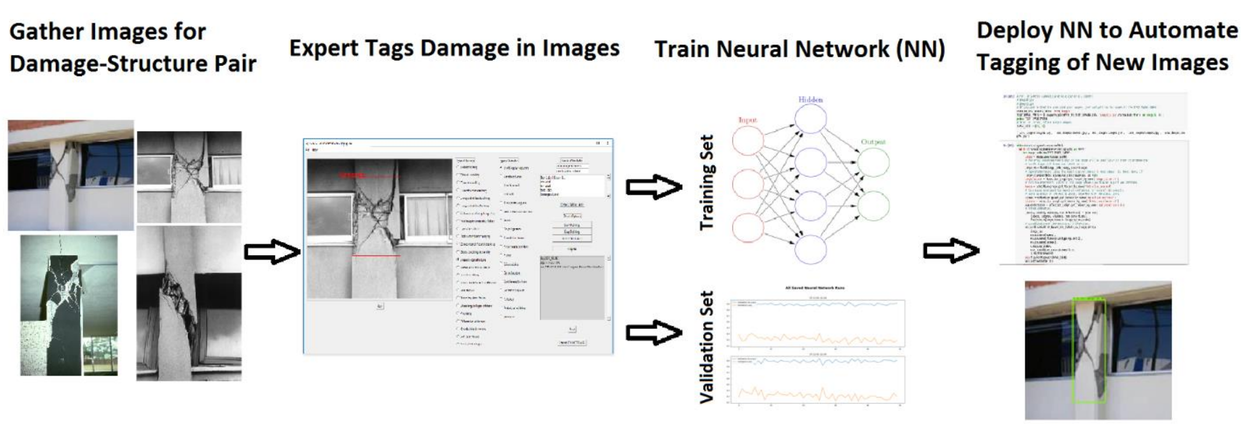

Figure 2. Flowchart for classifying reinforced concrete building damage.

The steps to develop a DL algorithm in damage-structure pair detection are as follows (refer to flowchart in Fig. 2):

1. Gather input images: While no specific number of images is necessary to train a DL algorithm, the general consensus is the more, the better. The authors' work thus far indicates ≥ 200 images per damage-structure pair yields accuracy similar to a human performance.

2. Manually tag the images: The authors created a software tool described in [9] used to draw a rectangular bounding box around every occurrence of the target damage-structure pair in each of the 200+ images. To store the rectangle coordinates and associated labels, DL frameworks require a specific format known as PASCAL VOC [10], which is one of the output types of the aforementioned software tool. The resulting set of manually tagged images becomes the "ground truth" with which to train the DL algorithm.

3. Divide the tagged images into two groups: Group #1 serves as the training set that will be supplied as the input to the algorithm in the learning process, and Group #2 is the validation set used to test that the algorithm is learning correctly. The authors chose TensorFlow [11] as the framework for building the DL systems since it is relatively easy to program and scales to multiple graphics processing units (GPUs), thus accelerating the training time.

4. Train: The larger research community has proven that specific DL network architectures (AlexNet and ResNet) [5, 7] can be generalized to solve different object classification problems by training on new input image data to adjust model hyper-parameters. The training starts with each neuron initialized to a certain weight, and subsequent iterations (epochs) consist of: (i) calculating the error/loss function with respect to the validation image set, (ii) modifying the neuron weights to correct for the calculated error using an optimization approach such as gradient descent, and (iii) re-training using the new weights. The iterations continue until the DL algorithm begins to converge and a satisfactory validation accuracy is achieved. Note that there are several parameters that can be adjusted to ensure the algorithm keeps learning rather than stalling in a local minimum. Additional details on the DL training process and related optimization approaches in can be found in [1].

5. Validate: Verify the DL algorithm is learning to identify the correct image features using the validation set created in Steps 2-3. Typically training and validation are done concurrently with only a small time offset to ensure the algorithm is learning at a desired pace.

6. Deploy: Once a satisfactory accuracy for detecting a specific damage-structure pair has been reached, the DL algorithm (contained in a Jupyter notebook) is tasked with identifying the same features in a completely new set of images. The trained NN for a given damage-structure pair, consisting on the weights for each neuron, can be now deployed to other users so to classify their own images.

Description of Graphical User Interface (GUI) Wrapper

A graphical user interface (GUI) wrapper was developed for the DL algorithm described in the previous section. This enables earthquake engineering/reconnaissance professionals to help further the field of computer vision for civil infrastructure, without having any programming or DL experience. The interface, shown in Fig. 3, only requires that users separate their images into folders based on the primary damage-structure type. The contents of these folders serve as the input with which the DL is trained to classify images into different categories (binning). The most significant outcomes include that non-programmers can independently: (i) prototype a new image classification DL model based on their unique set of photographs and damage-structure types, as well as (ii) review numerical/graphical performance metrics related to the accuracy in the training/validation of that new model. This tool was developed using Python [12], which allowed easy implementation of the Keras neural network library [13] running on top of TensorFlow [11].

When using the interface, it first necessary for the user to select the "Data" directory where the sub-folders containing the categorized input/training images are located. The names given to these sub-folders will be used as the category names for classification. After clicking the "Start NN" button, all images located in the "Data" directory are copied with respective sub-folders into a newly generated "Train" directory. Based on the validation percentage selected by the user, the appropriate number of validation images is calculated and random indices of that count are selected and moved to the "Validation" directory organized into sub-folders for the given categories. Finally, the "NNRuns" folder is generated which stores all the details of the neural network run.

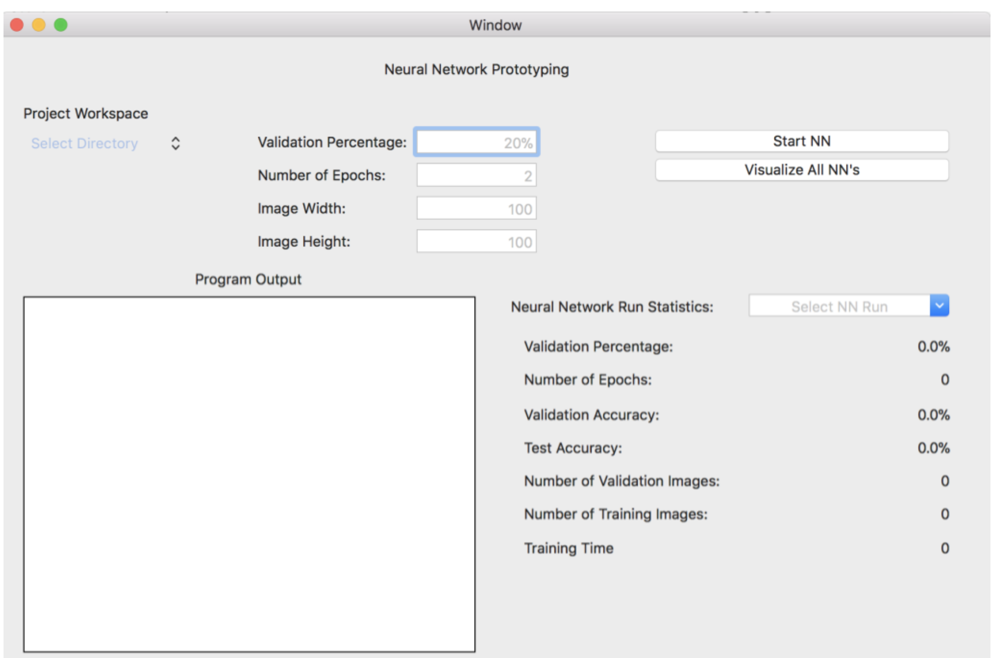

Figure 3. Graphical user interface (GUI) wrapper for the DL algorithm.

An example where the authors implemented the GUI wrapper was for the initial task of binary binning of building images as damage-no damage. A fifteen-layer neural network based on the Keras Sequential model [15] was constructed and compiled, and then the Keras image pre-processing function ImageDataGenerators [13] was utilized to augment the training image set by creating transformations (rotation, flip, shift, zoom, color alteration, etc.) of the original training and validation images. The resulting images were then scaled and manipulated to make the data uniform and to optimize the training process. Finally, the model was run and the output – including the validation accuracy, validation loss, total time, the loss, and accuracy – was stored in the "NNRuns" folder as a data object. The neural network weights are also saved to for use later to classify new images. Fig. 4 shows the numerical and graphical output available in the GUI wrapper. On the left is a summary of the neural network run statistics and on the right a plot of number of epochs versus validation accuracy (blue line) and validation loss (orange line).

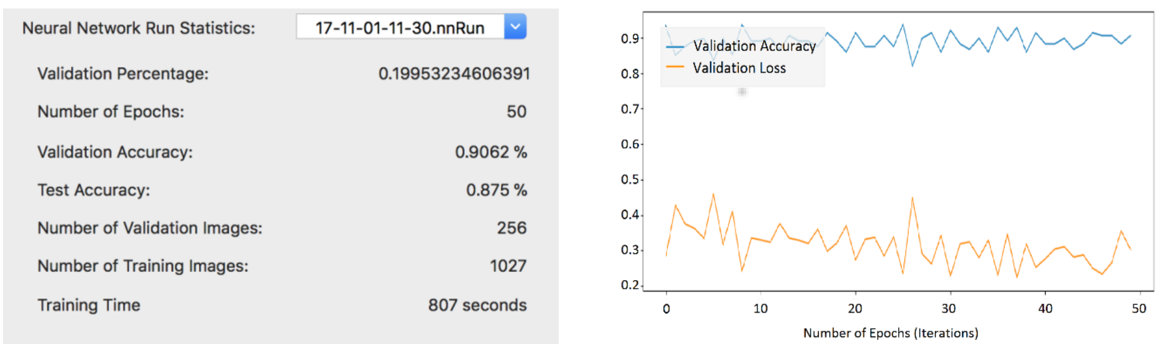

Figure 4. Neural network run data: (left) statistics, (right) graphic.

The plot presented in Fig. 4 is significant, since it summarizes how well the algorithm is learning and if the DL will converge with the intended training parameters. For instance, at high learning rates the validation loss will decay (trend toward 0.0) more rapidly; therefore, the desired response when training the DL is at least a small, but consistent, decay in validation loss. Conversely, the validation accuracy should increase (trend toward 1.0) over time without stalling at a local minimum. For both validation accuracy and loss, it is normal to see localized +/- peaks. This occurs since the updated weights for an epoch does not always improve the DL algorithm performance, but is usually corrected for on subsequent epochs.

Results of Different Deep Learning Approaches

For the binning classification approach, the authors utilized the aforementioned GUI wrapper. Around 200 training images were input to train the algorithm on binary binning of building images for damage versus no damage. The trained neural network was able to classify 1283 photos; despite being of a relatively generic and arbitrary network configuration, it was possible to achieve an accuracy of 88.3%. Fig. 5 includes a small subset of the images the algorithm correctly binned as damage versus no damage.

However, there is still room for improvement and currently the research team is working on implementing well-known classification neural networks including AlexNet [7] and ResNet [5] in the GUI wrapper tool to enable the user to determine the best fit for their specific input data. When complete, the true power of the GUI wrapper tool will lie in its versatility. It will be possible to use it on datasets with any number of classes and allow the user to customize the neural network. Therefore, the tool will be completely independent of the dataset, which means that it can be applied across a multitude of problems to assist in rapid prototyping of neural networks.

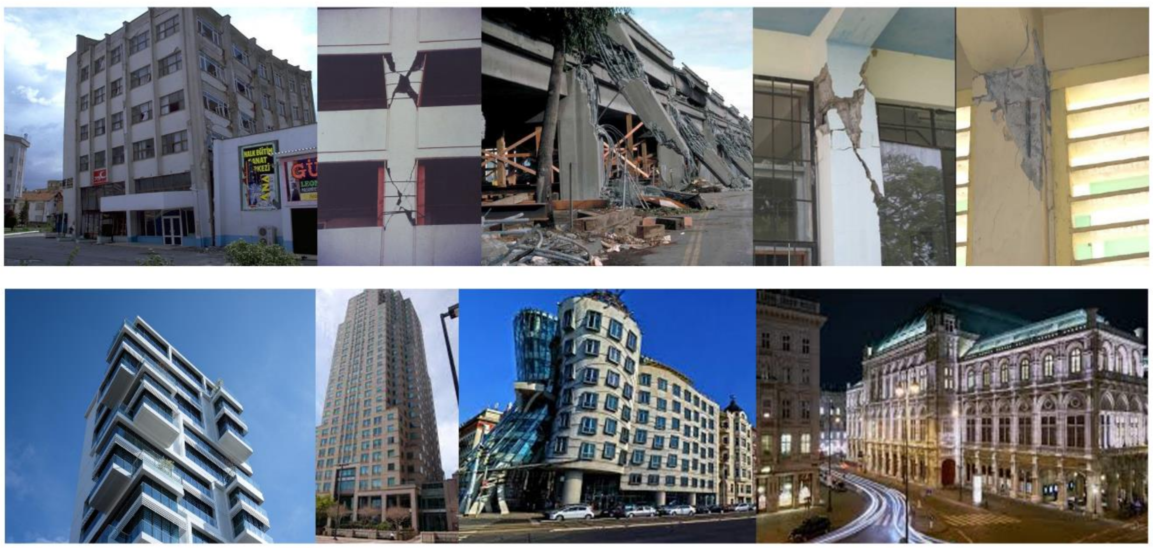

Figure 5. Binning classification results: (top) damage, (bottom) no damage.

For the bounding box classification approach, the authors implemented a Single Shot MultiBox Detector (SSD) [14] combined with the ResNet neural network [5] since this pairing has consistently shown classification robustness in various different computer vision tasks [16, 17]. The process involves training an individual damage-structural member pair at a time and requires a minimum of 200 images. To test the implementation, two sets of "ground truth" images were utilized: (i) damage versus no-damage (same as the training set for the binary binning classification described earlier), and (ii) the specific damage-structural member pair of shear damage to a short/captive column.

The accuracy of drawing bounding boxes around regions of buildings that exhibit damage (versus no damage) is 85%. This accuracy is slightly lower than for binning (88.3%) for the same classifiers, since drawing the bounding box around the damage location is more difficult than simply binary categorization of the image. Future work to increase accuracy for this task will involve using a larger and more diverse set of training images.

The DL algorithm's accuracy for drawing a bounding box around short/captive columns with shear damage is 77%. Fig. 6 presents a few examples of images the algorithm correctly tagged for this damage type. There are a few challenges with training for specific damage-structural member pairs that may explain the current level of accuracy. First, it is a time intensive process to find 200 high quality images from reputable sources that accurately represent the desired damage type and that only result from earthquake loading (not blast, windstorms, tsunami, etc.). Second, the research team is currently dependent on one expert tagging the images, while there needs to be multiple experts to serve as verification. Nevertheless, the current level of accuracy is rather promising and indicates that with a larger set of training images labeled by at least two experts, the DL algorithm's tagging performance would be comparable to a human expert.

Figure 6. Bounding box classification results: shear damage to short/captive column.

Conclusions and Future Work

The goal of the project is to develop a DL algorithm that will enable professional structural engineers to automatically label images for damage-structural member pairs commonly observed in civil infrastructure after earthquakes. Output images would have additional metadata that includes the damage-structural member types and locations in the images, which would enable large structural reconnaissance image repositories to become searchable using specific terms.

Current results show that a DL solution to classified damage/structure patterns is possible and as more images become available for training, the system it will be able to classify more complex tags. In the future, the authors intend to use approximately 500-1000 tagged images (using 2+ experts) for each damage-structural member pair, specifically in reinforced concrete buildings/infrastructure. The authors plan to make both the database of expertly tagged images and the DL algorithm for classification public so other structural engineers can contribute tagged reconnaissance images and develop their own NN model implementations.

The future of image recognition in civil infrastructure should be based on computer vision. This will not be a substitute for the knowledge of expert structural engineers; rather, it would facilitate their more rapid and targeted analysis of the important qualitative data found in reconnaissance images. By presenting experts with a filtered set of images, they would be able to concentrate their efforts in the way that DL algorithms are already helping radiologists to conduct focused analyses of MRI images for cancer detection [16, 17] by pre-selecting areas of interest.

Acknowledgments

The authors would like to acknowledge the support of California Polytechnic State University – San Luis Obispo via the CPConnect and Research, Scholarly, and Creative Activities (RCSA) Grant Programs.

References

1. Goodfellow I, Bengio Y, Courville A. Deep learning**.** *MIT Press* 2016. www.deeplearningbook.org.
2. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, Berg A, Fei-Fei L. ImageNet large scale visual recognition challenge. *Int'l Journal of Computer Vision* 2015, (**115**) 3, 211-252.
3. Brilakis I, German S, and Zhu Z. Visual pattern recognition models for remote sensing of civil infrastructure. *Journal of Computing in Civil Engineering* 2011, (**25**) 5, 388-393.
4. Feng C, Liu MY, Kao CC, Lee TY, Deep active learning for civil infrastructure defect detection classification. *ASCE International Workshop on Computing in Civil Engineering* 2017.
5. He Z, Zhang X, Ren X, Sun J. Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition* 2016.
6. Yeum, CM. Computer vision-based structural assessment exploiting large volumes of images. PhD Thesis, *Purdue University* 2016.
7. Krizhevsky A, Sutskever I, Hinton G. ImageNet classification with deep convolutional neural networks. *Advances in neural information processing systems* 2012.
8. Koch C, Doycheva K, Kasireddy V, Akinci B, Fieguth P. A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure. *Journal of Advanced Engineering Informatics* 2015, (**29**) 2, 196-210.
9. Behrouzi A, Pantoja M. "Photo Tagging Tool for Rapid and Detailed Post-Earthquake Structural Damage Identification." *Proceedings of the 11th National Conference in Earthquake Engineering* 2018.
10. Everingham M, Eslami SMA, Van Gool L, Williams CKI, Winn J, Zisserman A. **"**The PASCAL Visual Object Classes Challenge: a Retrospective" *Int'l Journal of Computer Vision* 2015, **111**(1), 98-136.
11. Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, Devin M, Ghemawat S, Irving G, Isard M, Kudlur M, Levenberg J, Monga R, Moore S, Murray DG, Steiner B, Tucker P, Vasudevan V, Warden P, Wicke M, Yu Y, Zheng X. TensorFlow: a system for large-scale machine learning. *12th USENIX Conference on Operating Systems Design and Implementation* 2016: 265-283.
12. Rossum G. Python library reference. Technical Report, *Centre for Mathematics and Computer Science Amsterdam, The Netherlands* 1995.
13. Chollet F, et al., Keras Repository. *Github*. https://github.com/fchollet/keras.
14. Liu W, Dragomir A, Dumitru E, Christian S, Scott R, Cheng-Yang F, Berg A. SSD: Single Shot MultiBox Detector. *Proceedings of the European Conference on Computer Vision* 2016.
15. Guide to the Sequential model. *Keras*. https://keras.io/getting-started/sequential-model-guide/.
16. Cheng JZ, Ni D, Chou YH, Qin J, Tiu CM, Chang YC, Huang CS, Shen D, Chen CM. Computer-aided diagnosis with deep learning architecture: applications to breast lesions in US images and pulmonary nodules in CT scans. *Scientific Reports* 2016, (**6**) 24454.
17. Esteva, A, Kuprel B, Novoa RA, Ko J, Swetter SM, Blau HM, Thrun S. Dermatologist-level classification of skin cancer with deep neural networks. *Nature* 2017. (542) 7639: 115-118.