17th World Conference on Earthquake Engineering, 17WCEE
Sendai, Japan - September 13th to 18th 2020

ADDING UNCERTANTY TO CONVOLUTIONAL NEURAL NETWORKS TO AUTOMATICALLY IDENTIFY STRUCTURAL DAMAGES

M. Pantoja(1), D. Fabris(2), and R. Kleinhenz(3)

(1) *Assistant Professor, Computer Science Department CalPoly San Luis Obispo, CA, USA, mpanto01@calpoly.edu*
(2) *Associate Professor, Mechanical Engineering Department, Santa Clara University, CA, USA, dfabris@scu.edu*
(3) *Lecturer, Applied Math Department, Santa Clara University, CA, USA, rkleinhenz@scu.edu*

Abstract

A critical task, after an earthquake, is determining the extent of damage to civil infrastructure. Nowadays every time there is a seismic event, large amounts of images are uploaded to the internet. In the past, these images would have been reviewed by trained volunteers or expert engineers to evaluate what kind of damage is shown. The manual review of large image-sets for assessing damage has shown to be inefficient and, in cases, error-prone.

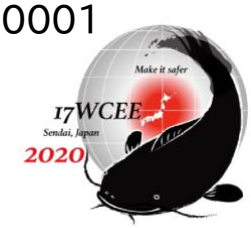
To speed up the process new computer vision algorithms are being proposed to automatically label the images for structural damages. Specifically, a deep learning approach using Convolutional Neural Networks (CNN) is proposed. Supervised CNN classify raw input data according to the patterns learned from an input training set. This training set is typically obtained by manually labeling images which can lead to uncertainties in the data. The level of expertise of the professionals labeling the training set sometimes varies widely or some of the images may not be clear and are difficult to label. This leads to data sets with pictures labeled differently by different experts or uncertainty in the experts' opinions.

Why measuring the uncertainty in CNN matters? Traditional CNN are trained to produce specific outcomes by optimizing a set of tunable parameters, the optimization is typically carried out using some form of gradient descent. For example, a CNN can be trained with labeled images of dogs and spiders. During the inference (deployment after training) the CNN will be able to automatically label new images of dogs and spiders. But what happens if during inference we feed the network the image of a cow? It will classify the image as a dog with high probability, since a CNN output predictive probability is just the probability with respect to the other labels, and a dog label is more probable then a spider. The CNN output predictive probabilities are often erroneously interpreted as model confidence. A CNN can be uncertain in its prediction even with a high SofMax probability output.

Uncertainty on the training set happens more frequently when the CNN task is to classify numerous labels with similar characteristics, as in our case when labeling damages on civil infrastructures after an earthquake. There are more than two hundred different label combinations and the experts labeling the sets frequently disagree on which one to use.

In this paper, we use probabilistic analysis to evaluate the uncertainty of the labels in the training set, these uncertainties will be used to "fuzzy" the output of the CNN. This will allow the computer classification to determine both the label and the uncertainty of the matching of that label. This way if a model returns a result with high uncertainty, we can decide to pass the input to a human for classification, instead of returning a completely wrong and potentially dangerous label. To prove the validity of our solutions we will test it on our dataset of images presenting shear damage-short column.

Keywords: Neural Networks, Belief Network, Probability Density Function, Structural Damage Classification.



1. Introduction

Advances in technology, particularly smart phones, have facilitated widespread image data collection and immediate dissemination following a seismic event. 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, timely, and efficient manner is challenging. There have been several efforts [2]-[7] to automate virtual post-earthquake reconnaissance activities by training a computer algorithm to classify images based on Convolutional Neural Networks.

In [5] Feng *et al.* attempts to address the challenge of having an adequately large, expertly tagged image training set. The research team developed a deep residual neural network 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. In [7] Yeum developed a convolutional neural network algorithm to recognize post-hazard structural damage in reconnaissance images. The damage classifications were collapse-no collapse (binary) and concrete spalling/flaking (bounding box). In [1]-[3] we developed a convolutional neural network for automatic damage classification on images after earthquakes.

Although the above research presents successes of CNN for earthquake image classification; still CNN can easily be fooled [17] giving high confidence predictions for unrecognizable images. The problem with uncertainty results on CNN has been previously studied. [9] Sun *et al.*, uses Bayesian learning to quantify posterior uncertainty on deep neural networks (DNN) models parameters; considering the matrix variate Gaussian to develop a scalable Bayesian outline inference algorithm by adopting a probabilistic backpropagation framework and stochastic gradient Markov Chain Monte Carlo (MCMC) on synthetic data. Kendall and Gal [10] analyzes the different kinds of uncertainty in the model and focus its work on the importance of adding aleatoric uncertainty (uncertainty due to statistical variations than cannot be predicted a priori) to the model, and proposes the use of Bayesian Neural Network for computer vision tasks improving 1-3% the model performance. Gal and Ghahramani [11] analyzes Neural Networks (NN) model certainty. In the paper they prove that the dropout layer can be used as a Bayesian approximation of a well-known probabilistic model, the Gaussian process. The paper uses these outputs to determine the model uncertainty and propose to pass the input to a human for classification if the output has high uncertainty. Deceus [12] proposes the use of belief functions to represent imprecise and or uncertain knowledge of class labels (soft labels) and proposed changes to common clustering algorithms to adapt to these types of labels, presenting result on synthetic data. Kendall *et al.* [20] presents a version of the segmentation algorithm SegNet that also outputs the uncertainty of the segmentation regions and is used for segmentation of street scenes. The authors provide as an output the uncertainty on each frame for the segmentation enabling users to decide on actions if the uncertainty is high. In general Bayesian Neural Networks (BNN) do not have fixed weights for the neurons but a distribution, quantifying the uncertainty in a NN which allows to find images for which the net is unsure of their prediction, but several experiments with BNN [14] show that they also provide a high level of certainty for out of distribution test data and require long training times. The study concludes that a Bayesian neural network with Monte Carlo dropout is too crude of an approximation to accurately capture the uncertainty information when dealing with image data.

In our previous paper, Pantoja *et al.* [1], we analyzed the problem in a different way than a BNN and instead of adding a probability distribution to the weights of the neurons we ask the individual expert for their certainty in labeling the images. Then through statistical and probabilistic analysis using

belief networks [15] we spread the CNN's predictive output over a range of values reflecting the expert's own self-certainty. The work presented in [1] mainly explains the mathematical methodology to evaluate the certainty of the model, the results were obtained using synthetic data. In this paper, we will evaluate the validity of the model on real data provided by the architecture department, specifically tagging damage after an earthquake on short columns. This way if a model returns a result with high uncertainty, we can decide to pass the input to a human for classification, instead of returning a completely wrong and potentially dangerous label. In Figure 1 we present an example of two different experts labeling the same image. It can be seen that there are clear differences.

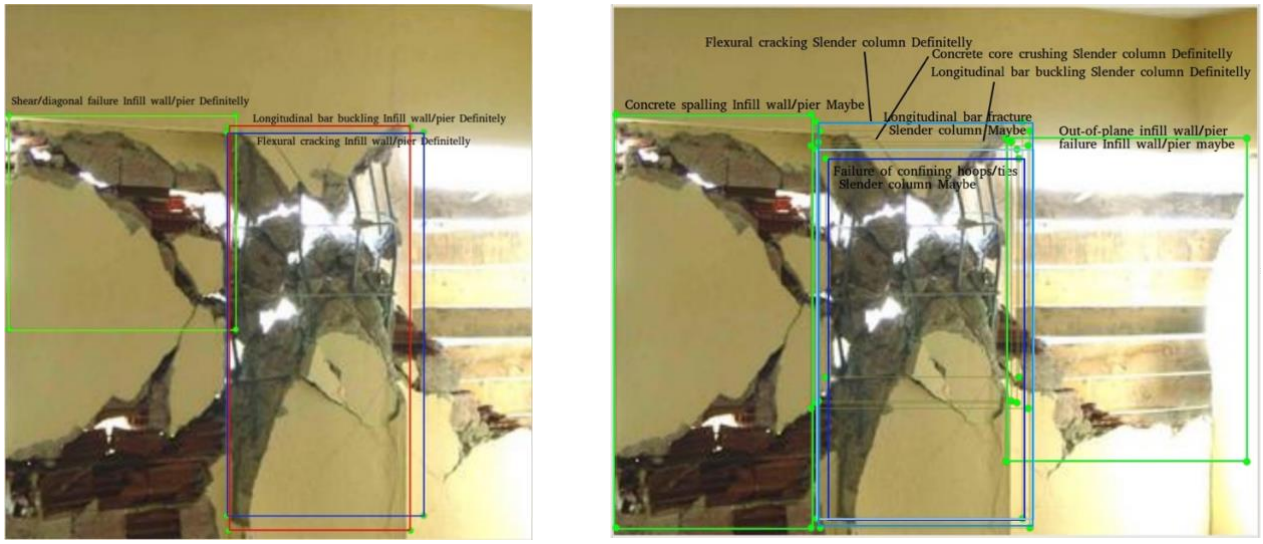


Figure 1. Two Experts Labeling Same Image

The article is structured as follows. In section 2 we give an overview of the implementation's details, first explaining what CNN for image classification is, then explaining certainty analysis. In section 3 we present our results, and sections 4 and 5 are analysis and conclusions.

2. Implementation

In this section, we explain how to evaluate the uncertainty of the labels created by the experts. We will start by giving a brief overview of Deep Learning algorithms and will follow with the mathematical theory behind our solution to evaluate the quality of the experts' labels.

2.1 CNN for Image classification

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. DL can model very complex inputs which allows researchers to shift from problem dependent feature extraction to a more general DL algorithm.

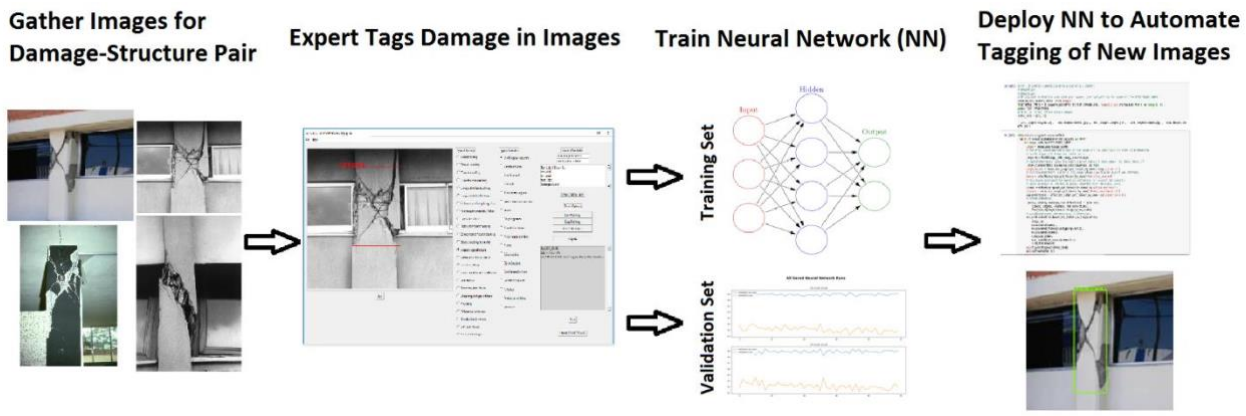


Figure 2. Flowchart for classifying reinforced concrete building damage.

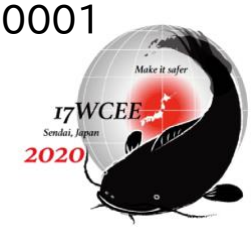
Development of CNNs involves two stages: (i) training on a set "ground truth" images that contain data known to be correctly classified and determining the optimal weight for the neurons, and (ii) deploying the NN using the learned weights from the training stage to classify a new image.

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

1. Gather input images.
2. Manually tag the images. To store the rectangle coordinates and associated labels using an image labeling tool (there are several on the internet that can serve this purpose). The resulting set of manually tagged images becomes the "ground truth".
3. Train: 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.
4. Deploy: Once a satisfactory accuracy for detecting a specific damage-structure pair has been reached, the NN can be used to identify the same features in a completely new set of images.

More detail about how to use DL for earthquake damage image classifications in our articles in [1]-[3].

2.2 Summary of the Uncertainty Analysis



Here we present a summary of the uncertainty analysis; a detailed explanation of the theory can be found in [1]. For the complete collection of photos evaluated by expert *E*, the success percentage is computed by forming the ratio of the total number of agreements found over all photographs examined by the expert to the total number of identifications (agreements, omissions, and additions) made by the expert. This ratio is the success proportion, $\tilde{p}$, and is given by

$$\tilde{p} = \frac{S}{M} = \frac{\sum_{\emptyset} c_{\emptyset}}{\sum_{\emptyset} e_{\emptyset} + t_{\emptyset} - c_{\emptyset}}$$

where *S* is the total number of agreements between *E* and the truth set and *M* is the total number of identifications made by *E* and the truth set (the sums are taken over all images examined by the expert *E*). For each image (call it $\emptyset$), let $e_{\emptyset}$, $t_{\emptyset}$, $c_{\emptyset}$ represent, respectively, the total number of labels identified by the expert, the total number of labels identified by the truth set, and the total number of labels common to both the expert and the truth set.

The proportion of successes $\tilde{p}$ is known to be a good estimate of the probability, *p*, that the expert *E* assigns a correct label to a photo. To estimate statistically how close $\tilde{p}$ is to *p*, a confidence interval, [*L*, *R*], is constructed around the parameter *p*. The natural confidence interval to be used is that for a proportion: well-known and given by the interval

$$[L, R] = [\ \tilde{p} - z_{\alpha/2}\sqrt{\tilde{p}(1-\tilde{p})/M}\ ,\ \tilde{p} + z_{\alpha/2}\sqrt{\tilde{p}(1-\tilde{p})/M}\ ]$$
$$= [\ \tilde{p}\left(1 - z_{\alpha/2}\sqrt{\frac{1-\tilde{p}}{\tilde{p}M}}\right)\ ,\ \tilde{p}\left(1 + z_{\alpha/2}\sqrt{\frac{1-\tilde{p}}{\tilde{p}M}}\right)\ ],$$

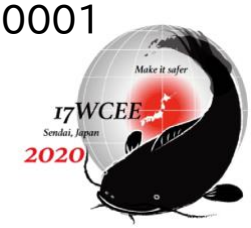
The confidence level, 1-α is the probability to be assigned (in this analysis) to the truth of the statement: $L \le p \le R$. The base of the triangular distribution function for V (the certitude that image phi possesses label L) is found by scaling the interval [*L*, *R*] by *V** (the expert's self-assessment). That is the endpoints of the base of the triangle are given by $a = V^*L$ and $b = V^*R$. It is also desired to keep $0 < V^*L < V^*R < 1$. Therefore, set $a = 0$ if $V^*L < 0$, set $b = 1$ if $1 < V^*R$. In summary:

$$a = \max\ (0, V^*\tilde{p}\left(1 - z_{\alpha/2}\sqrt{\frac{1-\tilde{p}}{\tilde{p}M}}\right))$$

$$b = \min\ (1, V^*\tilde{p}\left(1 + z_{\alpha/2}\sqrt{\frac{1-\tilde{p}}{\tilde{p}M}}\right)) .$$

To make this triangular function into a density function, the area under this triangle must equal one. Therefore, the height of the triangle must be $2/(b-a)$. Outside of this triangle the density function is zero.

Let's explain all this with a numerical example were we have two experts evaluate the same set of photos. Table 1 presents a numerical example of how to evaluate the quality of these two experts assigning labels to two different photos. The label name indicates a damage/structure type and its location, for example: "shear flexture Short column 4 100 190 300", where the four numbers at the end indicate the x-y coordinates of the rectangle in the photo that contains damage of the type shear



flexure. This indicates that different damages names with same locations will be consider different labels. The data in Table 1 represent a case in which two experts have evaluated two photographs and attached up to three labels (called *A*, *B*, and *C*) to each. Out of these values it is now possible to determine the shape of each of the certainty functions, a summary of the calculation can be found in Table 2. The certainty functions are denoted as

$$f_{L,E,\Phi}(l) \quad .$$

To ease the notation here this form will be abbreviated to $f_{L,E,\Phi}$ corresponding to identifiers *l*: label, Φ: photograph, and *E*: expert. Note that for these intervals, a confidence level of 0.05 was assumed. The first rows in Table I are the labels assigned by experts to a specific photo. We compare them to the ground truth value found on the third column of same Table I. The next rows are the calculations for the width and height of the certainty density function. Table II shows the calculations for density functions, the graph of the values in Table II are represented in Figure 3.

Table I. Representative Classification

Expert 1	Expert 2	Ground Truth
Photo 1	Photo 1	Photo 1
label A Maybe: 0.08	label A Definitely: 0.92	label A Definitely: 0.92
label B Certain: 0.64	label B Certain: 0.64	label B Definitely: 0.92
label C Maybe: 0.08		
$e_\Phi = 3 \quad c_\Phi = 2$	$e_\Phi = 2 \quad c_\Phi = 2$	$t_\Phi = 2$
$M_{1,1} = e_\Phi + t_\Phi - c_\Phi = 3$	$M_{2,1} = e_\Phi + t_\Phi - c_\Phi = 2$	
Photo 2	Photo 2	Photo 2
label D Certain:0.64	label D Definitely:0.92	label D Definitely:0.92
$e_\Phi = 1 \quad c_\Phi = 1$	$e_\Phi = 1 \quad c_\Phi = 1$	$t_\Phi = 1$
$M_{1,2} = 1$	$M_{2,2} = 1$	
$\tilde{p}_1 = 3/4$	$\tilde{p}_2 = 1$	

Table II. Distribution Functions

(l, e, ϕ)	a = **L**eft	V$^*\tilde{p}$ = **C**enter	b = **R**ight
$(A, 1, 1)$	0.026	0.08	0.094
$(B, 1, 1)$	0.208	0.64	0.751
$(C, 1, 1)$	0.026	0.08	0.094
$(D, 1, 2)$	0.208	0.64	0.751
$(A, 2, 1)$	0.92	0.92	0.92
$(B, 2, 1)$	0.64	0.64	0.64
$(D, 2, 2)$	0.92	0.92	0.92

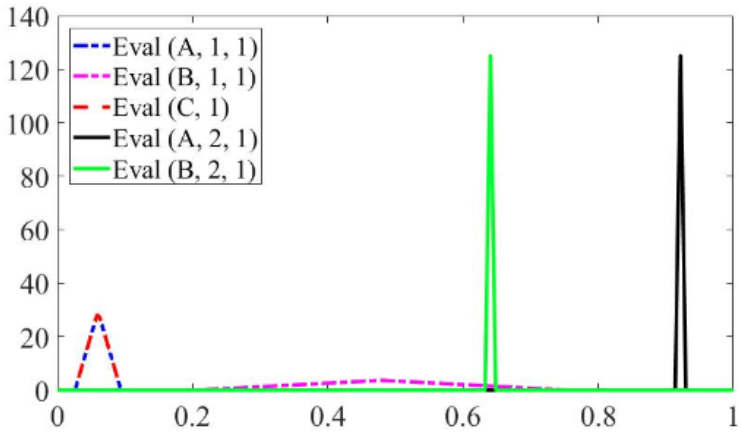


Figure 3. Experts Density Function Shapes (for Photo 1):$f_{l,e,1}$

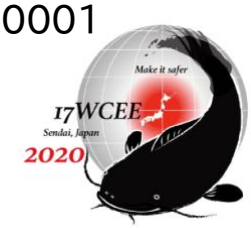
The certainty functions $f_{l,e,1}$ obtained from the values for Photo 1 (stated in Table 1) are represented in Figure 3. It can be seen that the certainty functions associated with expert 1 are triangular and wider while the certainty functions associated with expert 2 are impulsive. This is due to the fact that expert 1 made several mistakes in assigning labels (compared to the truth set) while expert 2 is not only certain about the labels but has also made correct assignment.

3. Results

3.1 Automatic Detection of "Shear Damage"

To test the implementation, a set of "ground truth" images were utilized the specific damage-structural member pair of shear damage to a short/captive column. The DL algorithm's accuracy for drawing a bounding box around short/captive columns with shear damage is 77% with an input training set of only 200 "ground truth" images. Figure 6 presents a few examples of images the algorithm correctly tagged for this damage type. The low number of images on the ground truth training set is the main reason for the relatively low accuracy of the model. 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. More details about this results can be found in [2].

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



3.2 Automatic Detection "Shear Damage" with uncertainty analysis

We did run algorithm as in 3.1 but this time we added also the confidence information of the experts labeling the training set. An example of the data is presented in Table III. Column "DL Output" contains the value for the softmax output layer of the DL model (explained in Section 2.1, 3.1); Columns "Expert 1" and "Expert 2" contain the certainty self-evaluation of the expert 1 and 2 respectively, and column "Ground Truth" contain the value for the ground truth (the real value of the label confirmed by infield evaluation); for this sample the ground truth confirms the exitance of the label on all images.

Applying calculations described in section 2.1 the $\tilde{p}$ for Expert 1 is 0.6 and for Expert 2 is 0.3; so it is very clear that Expert 1 is more often correct and should be given in general more weight on the training set since the quality of its labels is higher.

Table III. Input for the Certainty Analysis

Image Number	DL Output	Expert 1	Expert 2	Ground Truth
1	0.8	Definitely	Probably	yes
2	0.9	Definitely	Probably	yes
3	0.9	Definitely	Definitely	yes
4	0.9	Probably	maybe	yes
5	0.8	Definitely	definitely	yes
6	0.6	Probably	Maybe	yes
7	0.8	Definitely	most probably	yes
8	0.9	Probably	definitely	yes
9	0.8	Definitely	probably	yes
10	0.7	Definitely	Maybe	yes
11	0.8	Probably	Maybe	yes
12	0.6	Maybe	Maybe	yes

Evaluating the quality of the expert is very important but on this paper we wanted to go further and use probabilistic analysis to evaluate the uncertainty of the labels in the training set, these uncertainties will be used to "fuzzy" the output of the CNN. This will allow the computer classification to determine both the label and the uncertainty of the matching of that label. In Table III, we know that column "DL Output" presents the output of the Softmax, we also know the label "Shear Flexture Short Column" is correct since the ground truth confirms it, therefore in theory all softmax outputs should be high (closer to 1.0). If the experts are not certain of the label these softmax value should be "fuzzied" (given a range) out. For example, for Image 12 (Row 12 on Table III) the DL output is 0.6, the ground truth confirms the label is there, Expert1 certainty is only "Probably" and Expert 2 is "Maybe"; in both cases the certainty should have been "Definitely". These uncertainty in both experts should fuzzy out the DL Output softmax. The results of the certainty analysis can be seen in Table IV and in Figure 7. Figure 7 present the output we create for the DL after applying the uncertainty analysis in Section 2.1 as expected the

output is of the DL is made wider lowering the value for the Softmax since the experts where not so sure of the label.

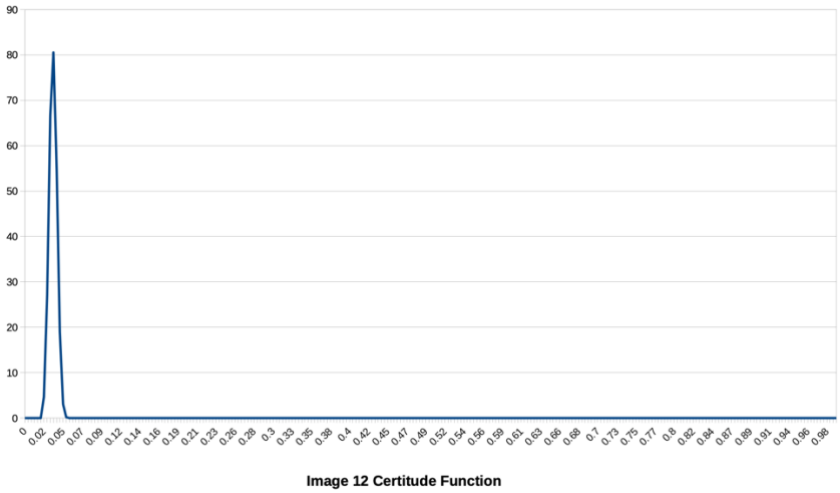


Figure 7. Certitude Function

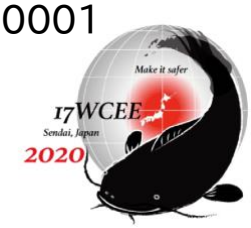
Table IV. Image 12 Certitude Function Parameters

Image #	NN Prob	Expert 1	Expert 2	Truth	Expert 1 Total Quality	Expert 2 Total Quality
12	0.6	Maybe	Maybe	yes	0.6	0.32
			Mean	0.03466		
Calculated from Certitude Function :			Variance	2.23E-05		
			St Dev	0.004719		

4. Conclusions

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 labels.

Since we do greatly need experts to classify the training images, we need to evaluate and validate the experts before training algorithms. In this work we present a methodology that works on adding the certainty to the data, providing a more accurate output of the DL since now is not a yes/no the damage is on the image but also a certainty of the classification.



5. Further Work

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. The classification of images taken after earthquakes is a very complex task and we need more data properly labeled to be able to provide accurate classifications, in the future we plan to expand our work to different labels and experts classifying the labels.

6. References

[1] Pantoja M, Kleinhenz R., and Fabris D. (2019): Adding Probabilistic Certainty to Improve Performance of Convolutional Neural Networks, CARLA High Performance Computer Conference Latin America , September 25-28, Turrialba, Costa Rica.

[2] Patterson B, Leone G, Pantoja M, Behrouzi A (2018): Deep learning for automated image classification of seismic damage to built infrastructure. *Proceedings of the 11th National Conference in Earthquake Engineering*, Los Angeles, CA, USA.

[3] Pantoja M, Fabris D., Behrouzi A (2018): Deep Learning Basic Overview. *Concrete International Magazine*, **40** (9), 35-41.

[4] 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.

[5] Feng C, Liu MY, Kao CC, Lee TY, (2017): Deep active learning for civil infrastructure defect detection classification. *ASCE International Workshop on Computing in Civil Engineering*, Seattle, USA.

[6] He Z, Zhang X, Ren X, Sun J. (2015): Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, Boston, USA.

[7] Yeum, CM. Computer vision-based structural assessment exploiting large volumes of images. PhD Thesis, *Purdue University* 2016.

[8] Tesla Crash Preliminary Report US department of transportation NHTSA PE 16-007 (2017).

[9] Sun S, Chen C, and Carin L (2017): Learning Structured Weight Uncertainty in Bayesian Neural Networks. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)* Fort Lauderdale, Florida, USA. JMLR: W&CP volume 54

[10] Kendall A, and Gal Y (2017): What uncertainties do we need in Bayesian Deep Learning for Computer Vision. *NIPS* https://arxiv.org/abs/1703.04977

[11] Gal Y, and Ghahramani Z. (2016): Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. *Proceedings of the 33rd International Conference on Machine Learning* PMLR 48:1050-1059.

[12] Deceus T. (2009): Handling Imprecise and Uncertain Class Labels in Classification and Clustering. Bayesian Deep *Learning COST Action IC 0702 Working group C*, Mallorca, Spain.

[13] Gal Y (2015): What my Deep Learning model Doesn't know. Personal blog.

[14] Hackerman D (1992): The Certainty-Factor Model. Encyclopedia of Artificial Intelligence Second Edition Wiley, New York pp. 131-138

[15] Pearl J (1998): Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann San Mateo CA

[16] Klir G, Yuan B (1996): Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems Selected Papers, Advances in Fuzzy Systems Applications and Theory Vol 6 World Scientific.

[17] Knuth D (1998): The Art of Computer Programming, Vol 2, Section 4.3.3, pp 290-295

[18] Press WH, Teukolsky SA, Vetterling WT, and Flannery BP, (1986): Numerical Recipes in C, Section 8.10, pp 329-343.

[19] Silver, D., Schrittwieser, J., Simonyan, K. *et al.* Mastering the game of Go without human knowledge. *Nature* **550,** 354–359 (2017). https://doi.org/10.1038/nature24270
[20] Kendall A, Badrinarayanan V, and Cipolla R (2015): Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding. *CoRR*, http://arxiv.org/abs/1511.02680,
[21] Weideman H. (2016): Quantifying Uncertainty in Neural Networks. https://hjweide.github.io/quantifying-uncertainty-in-neural-networks .
[22] Avis D, and Fukuda K (1992): A Pivoting Algorithm for Convex Hulls and Vertex Enumeration of Arrangements and Polyhedra. Discrete & Computational Geometry, **8** (3), 295–313
[23] Liu W, Dragomir A, Dumitru E, Christian S, Scott R, Cheng-Yang F, Berg (2016): SSD Single Shot MultiBox Detector. *Proceedings of the European Conference on Computer Vision*.
[24] github:https://github.com/mpantoja314/ImageTagVER

