DEEP CASCADED NEURAL NETWORKS FOR AUTOMATIC DETECTION OF STRUCTURAL DAMAGE AND CRACKS FROM IMAGES

Yongsheng Bai[1*], Bing Zha[1], Halil Sezen[1], Alper Yilmaz[1]
[1] Dept. of Civil, Environmental and Geodetic Engineering, The Ohio State University, 2070 Neil Avenue, Columbus, Ohio, USA – (bai.426, zha.44, sezen.1, yilmaz.15)@osu.edu

Commission II, WG II/5

KEY WORDS: Deep learning, structural damage detection, cracking detection, cracking localization, ResNet, U-Net

ABSTRACT:

In this paper, two different convolutional neural networks (CNNs) are applied on images for automated structural damage detection (SDD) in earthquake damaged structures and cracking localization (e.g., detection of cracks, their widths and distributions) at various scales, such as pixel level, object level, and structural level. The proposed method has two main steps: 1) diagnosis, and 2) localization of cracking or other damage. At first a residual CNN with transfer learning is employed to classify the damage in the structures and structural components. This step performs damage detection using two public datasets. The second step uses another CNN with U-Net structure to locate the cracking on low resolution images. The implementations using public and self-collected datasets show promising performance for a problem that had remained a challenge in the structure engineering field for a long time and indicate that the proposed approach can perform detection and localization of structural damage with an acceptable accuracy.

1. INTRODUCTION

1.1 Introduction

Automatic Structural Damage Detection (SDD) requires advanced technologies to determine the level of damage experienced by a structure and to evaluate the service life, integrity, stability and safety of the structures, for example, after major events such as hurricanes and earthquakes. Remote or non-destructive techniques, such as high definition cameras, can be used to assess and evaluate the state of the structure. Typically, the damage condition of the structure is evaluated by human experts during field inspection (Yang et al., 2017). Field inspections may be risky, unsafe or very expensive to conduct, especially after a major disaster such as an earthquake or extreme wind event. On the other hand, with the development of optical and robotics industry, vision-based technologies are becoming more viable and competitive. Many researchers and engineers have started to look into this technology for field inspection to detect structural damage and monitor the health of the structure, i.e., damage progression in the structure (Spencer et al., 2019).

With recent advances and widespread use of high definition cameras, drones, and robots, it is imperative to develop a model that can automatically detect and classify various types of damage accurately using computer vision methods (Spencer et al. 2019, Gao and Mosalam, 2018). Robots can be trained to recognize structural damage automatically so that the cost and time can be reduced dramatically for field inspections.

1.2 Problem Statement

In this research, a deep learning method is adopted to detect several structural failures with PEER Hub ImageNet (Phi-Net) dataset (Gao and Mosalam, 2020). The dataset includes different damage levels and types, collapse, spalling, and scene classification. The work presented in this paper has similarities to those of Yeum et al. (2018) and Gao and Mosalam (2018) with a focus on how deep learning models can accurately identify the damage on structures without human intervention. However, these cited methods cannot identify the locations of damages, while human experts can easily identify the structural damage by checking the images manually. This paper proposes an approach to perform both detection and localization of one of the damage, cracking, automatically in this way: First, a ResNet model (He et al., 2016, Zha et al. 2019) is used to classify the cracking, then the U-Net model (Ronneberger et al., 2015, Flôr, 2019) is employed to mask cracks after training, so that the location of the cracking can be determined.

It is necessary to consider the effect of scale when the cracks are located on structures since they may look totally different in images taken at different distances. Structural damage manifest themselves differently at various levels: pixel level, object level and structural level (Figure 1). In pixel-level images, typically structural components are zoomed in and partially captured. They fully appear in object-level images, so columns, beams and walls can be recognized. On the other hand, an entire building or bridge can be seen in structural-level images. The typical cracks in pixel-level images are wide and deep while in object-level ones they look long and narrow. Cracks also appear in structural-level images, however, there are more other objects and less visibility as scale increases. These characteristics are the reasons why we select typical images at different scales before labeling the samples and training the proposed model to automatically locate the cracks.

(a) pixel level (b) object level (c) structural level
Figure 1. Three scene levels (scales) in the proposed model

The paper is organized as follows: Section 2 provides a brief review of related research. Section 3 introduces a deep learning method, the ResNet, for classification. Section 4 shows the implementation with the U-Net to segment and locate the cracks.

* Corresponding author

Section 5 discusses the problem of crack localization at different scales, and Section 6 provides concluding remarks.

2. RELATED PREVIOUS RESEARCH

2.1 Deep CNNs for Structural Damage Detection and Cracking Localization

There has recently been increasing number of publications on structural damage detection. Yeum et al. (2018) use AlexNet to classify and identify the structural damages in post-event buildings with large scale images. Hoskere et al. (2017) try an experiment with 23-layer ResNet and 9-layer Visual Geometry Group (VGG) networks to classify and segment 7 classes of structural damage. which include cracks, spalling, exposed reinforcements, corrosion, fatigue cracks, asphalt cracks, and no damage. Ali et al. (2019) introduce Faster R-CNN (Faster Region Convolutional Neural Networks) into defects detection in historical masonry buildings with high resolution images. Kong and Li (2018) describe an application that detects and tracks the propagation of cracks in a steel girder with a video stream. Atha and Jahanshahi (2018) explain the different effects when they use two algorithms of CNNs (VGG16 and ZF Net) in detecting metallic corrosion.

Gao and Mosalam (2020) started the Phi-Net Challenge for collecting pictures of building structural failures, which is used as a dataset in our work. There are eight tasks in this dataset: 1) scene level: it can be used to detect cracks at pixel level and identify concrete spalling on structural components and collapse of buildings and bridges at object level and structural level, while spalling means the concrete cover of the steel reinforcements is split from the base; 2) damaged or undamaged state; 3) spalling or Non-spalling; 4) material type: steel and others; 5) collapse mode: this task distinguishes global collapse, partial collapse and non-collapse of structures; 6) component type: there are four types, including beams, columns, walls and others; 7) damage level: no damage, minor damage, moderate damage and heavy damage; 8) damage type: it can identify four types of structural member failure, including no damage, flexural damage, shear damage and combined damage (Gao and Mosalam, 2018，Gao and Mosalam, 2020). The cracks position, orientation and shape are good indicators to determine whether moments, shear forces or both on the structural components cause material failures, which could be called as flexural damage, shear damage or combined damage. There are totally 36,413 images with various scales in this dataset. The extended framework of Phi-Net is shown in Figure 2.

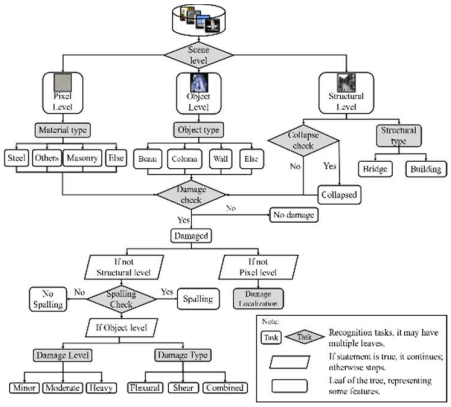

Figure 2. Hierarchy of the extended Phi-Net framework (Gao and Mosalam, 2020)

Zhang et al. (2018) propose an improved CNN (Convolutional Neural Network) for autonomous detection of pavement cracks at the pixel level. Liu et al. (2019b) demonstrate the application with U-Net to segment the crack on concrete structures, and their experiment shows the proposed network outperform the CNN which was used by Cha et al. (2017). Dung and Anh (2019) also use Fully Convolutional Network to localize the cracks on the concrete surface, and Liu et al. (2019a) implement Deep-Crack, which adopts an extended Fully Convolutional Networks (FCN) and a Deeply-Supervised Nets (DSN), to detect pixel-wise cracks. These methods to segment the cracks are based on pixel level and less useful in SDD for structural engineers.

2.2 COCO Like Data Labeling

In our work, we curated a dataset similar to Common Objects in Context (COCO) and used it for training the pipeline. COCO is a large-scale object detection, segmentation, and captioning dataset (cocodata-set.org/home). COCO has several features: object segmentation, recognition in context, super-pixel segmentation, 330K images (>200K labeled), 1.5 million object instances, 80 object categories, 91 stuff categories, five captions per image.

The COCO dataset does not contain structural damage, and there are only a few open sources for cracking segmentation at hand. For this reason, we curated images and created cracking dataset. The images in our dataset are at various scales and are resized using the tool referred to as the COCO Annotator (Brooks, 2019) to label cracks for training. Some examples from this process are shown in Figure 3. In these labeled images, cracks are in yellow and back-ground is in purple. Image size of all the training and labeling images is 256×256. We will make this dataset available to other researchers when the paper is published.

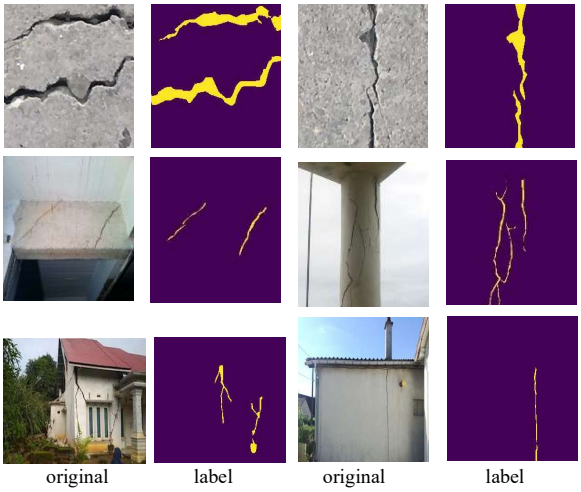

Figure 3. Some examples of training data

3. FIRST NETWORK FOR CLASSIFICATION: RESIDUAL NEURAL NETWORK (RESNET)

In this section, the architecture of the classification network, which is coded as a ResNet, is briefly introduced. Then, we have tested our implementation of ResNet on two datasets, Phi-Net and an open-source dataset of concrete surface cracks which are collected from several buildings at Middle East Technical University (Özgenel, 2018) for analyzing its performance.

3.1 Architecture of Our ResNet

He et al. (2016) introduced the architecture of ResNet. The ResNet is built upon each layer which learns a residual function $F(x) := H(x) - x$ (see Figure 4, $H(x)$ is an underlying mapping of x) and x itself as the equation $y = F(x) + x$, instead of a direct

mapping of $y = H(x)$ as done in many other CNNs. In contrast to regular neural networks which gets saturated and suffers a performance degradation as the network depth increases, ResNet and its residual learning strategy can overcome vanishing and exploding gradients when some connections between layers are skipped. Therefore, ResNet makes deep networks possible and show higher accuracy on the tasks like image recognition. Zha et al. (2019) provides two reasons for this better performance: from the perspective of mathematics, it is reasonable to set the residuals to zero than fit to an identity mapping x by stack of non-linear layers if the identity mapping is optimal, since the operation for $F(x) = 0$ instead of $F(x) = x$ is much easier in neural networks. Second, from an intuition perspective, the whole hierarchical feature combinations can be optimized with skipping connections and fewer feature compositions may better represent the objects in various layers.

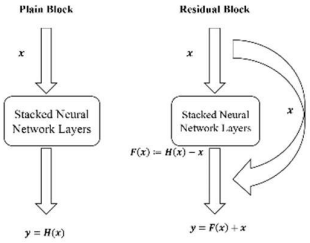

Figure 4. Block of plain net and Residual net

In this paper, a ResNet with 152 layers is used because its performance is better than others. The steps are as follows: we implement 152 layers of ConvNet layers and FC (Fully Convolutional) layers, and three different filter sizes on ConvNet layers, 1×1, 3×3 and 7×7 are employed. Different filter size of convolution generates different feature representations for different scales. The 7×7 convolution filter is on top of the network and follows four blocks of convolution combination, each of which consists of two 1×1 convolutions and one 3×3.

3.2 Performance on Phi-Net Dataset

For testing the network performance, the hyperparameters are defined as: learning rate is 0.001 and momentum is 0.9; the loss function is cross-entropy for classification problem. In addition, 40 min-batches are defined to maximize GPU usage while the total number of epochs is set as 100. The training and testing for the model are executed with NVIDIA GeForce GTX 2080 Super.

For analyzing performance, we used accuracy that represents the percentage of the correctly classified images:

$$Accuracy = \frac{\sum(True\ Positive)}{N} \quad (1)$$

where N = total number of samples.
Testing results are shown in the Table 1.

Table 1. Classification Results on Testing Data

Detection Tasks	Number of Classes	Image Statistics			Test Acc (ResNet)
		Train	Validation	Test	
Scene Classification	3	13,939	3,485	4,356	93.8%
Damage Check	2	4,730	1,183	1,479	81.9%
Spalling Condition	2	2,635	659	824	79.6%
Material Type	2	3,470	867	1,085	99.5%
Collapse Check	3	2,105	527	658	63.1%
Component Type	4	2,104	526	658	71.7%
Damage Level	4	2,105	527	658	67.8%
Damage Type	4	2,105	527	658	67.5%

Our ResNet model can classify scene levels and material types well as shown in Table 1, whereas its accuracy on checking collapse and identifying damage levels and types is not high, but it is acceptable for current data collection.

3.3 Performance on Dataset of Concrete Surface Cracks

There are two tests on this dataset, one for scene classification and the other for identifying cracking types. The total number of images in the datasets is 40,000, and only half of the images have surface cracks. Image size of this dataset is 227×227, and they are resized to 224×224.

3.3.1 Scene classification After the dataset is tested with the proposed ResNet model, all the images were classified as pixel level cracks in Task 1, suggesting that our model performed well.

3.3.2 Crack type: There are four types of damages (related to cracking types) in Task 8: non-damage, flexural damage, shear damage, combined damage. A test is performed on this dataset after training our model with the Phi-Net dataset. Table 2 shows the result. The model can achieve 91.21% accuracy on finding the cracks in this image dataset.

Table 2. Classification of Cracking and Noncracking

Cracking Type	Number of images		
	Noncracking	Cracking	Total
Non damage	19,184	2,699	21,883
Flexural damage	228	4,490	4,718
Shear damage	84	7,702	7,786
Combined damage	504	5,109	5,613
Accuracy	95.92%	86.505%	91.21%

4. SECOND NETWORK FOR DETECTION: U-NET

After the first network is used to identify these images with various cracks on the structures or structural components, U-Net is employed to locate them. It has been successfully applied on biomedical image segmentation (Ronneberger et al. 2015). As shown in Figure 5, the U-Net architecture is a symmetric structure, the left part consists of several convolutional layers while the right side is made of up-sampling layers, or they can be called encoder and decoder respectively. But the features extracted from the same size convolutional layers are concatenated with corresponding up-sampling layers, thus these high or low-level feature maps can be kept and inherited by decoder to get more precise segmentation.

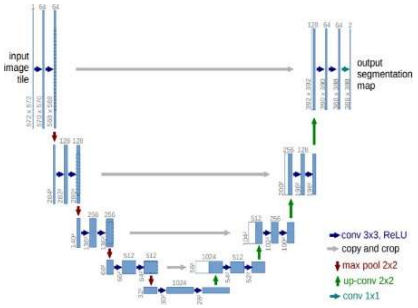

Figure 5. U-Net architecture (Ronneberger et al., 2015)

The developed U-Net model has a similar architecture as shown in Figure 5 (Flôr, 2019). In this paper, data augmentation is applied prior to training so that the prediction can be more accurate with limited data. Thus, there are 1,000 images with pixel-level cracks in our U-Net model, and 853 images for object-level and structural-level cracks. Both datasets are separately trained on the same computer with GPU, and learning rate is 0.0001 and binary cross-entropy is assigned as the loss function.

We define the pixel-level cracks as partially or incompletely cracking in images, in which the structural components such as walls, columns, beams, slab and nonstructural components like partition walls and decoration layers are also not included completely. In addition, most of them are also 2D or planar cracks

on these components. Contrast to this, those cracks on structural components or an entire structure are labeled as object-level and structural-level cracks.

In the following figures to demonstrate the results of our implementations, white color in prediction images means cracking while background is in black, and cracks are marked in red while light blue color is the background in the overlaid images.

4.1 Locating the Pixel-level Cracks on 2D Images

4.1.1 Concrete surface cracks

For this test, 1,000 out of 20,000 images with cracks in the dataset of concrete surface crack are randomly selected and labeled through COCO Annotator. Upon training, the method achieved an accuracy 96.48% for validation set. The testing data are the remaining 19,000 images of this dataset.

Some results are presented in Figure 6. As can be observed that the U-Net structure was able to successfully predict and mask the cracks on concrete surfaces, or it almost reproduces the same pattern of real cracks through learning. Overall, there were no failure cases in locating the cracks in this test, but few examples as shown in Figure 7 indicate that there were some noise in final labels. The percentage of these cases in this test is less than 5%. The problem can be attributed to the limited number of images in the training set. We believe that the accuracy would improve if the training data size is increased.

4.1.2 Pixel-level data in Phi-Net

The above trained U-Net model is implemented on pixel-level data from Phi-Net, which includes more scenarios of cracking, which are not only confined on concrete surface but also on the surface of masonry and decoration layers. There is a total of 4,661 images, but cracking and non-cracking can be successfully detected in 2,819 images. Since they are not identified or wrongly marked in 1,842 images, the accuracy for our U-Net model is 60.48% while there are a few similar images from this training dataset, with respect to the scale of testing images varies in a flexible range but uniform in the training dataset. In addition, this method is also an end-to-end test for classifying and segmenting cracks at pixel level. These results show that the U-Net with less training data for detecting pixel-level cracking can work well. A higher accuracy can be achieved if a larger number of similar images can be fed into the dataset in the future.

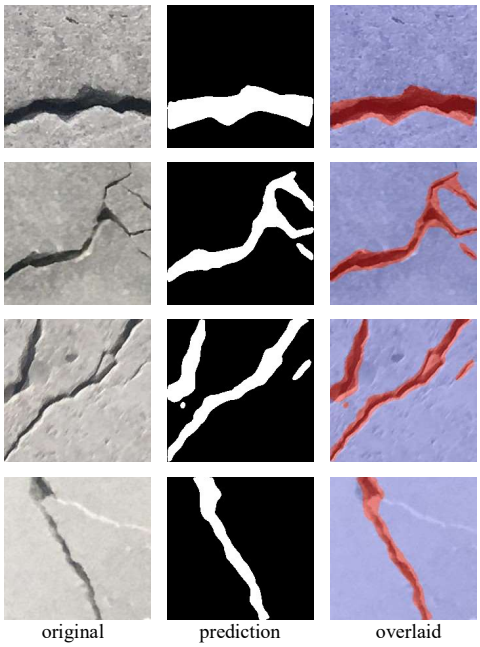

Figure 6. Some examples of normal testing results

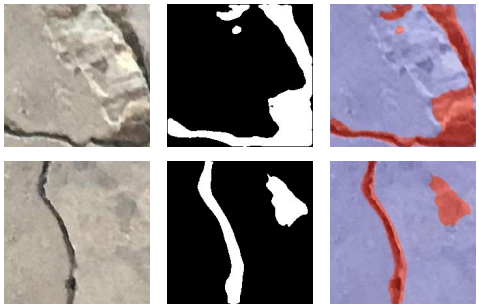

Figure 7. Some examples of testing results with noise

Figures 8 and 9 show some examples of accurate and false predictions respectively. Figure 8 shows that the model can provide a very precise prediction for the location of the cracks in pixel-level data whereas the surface is not the same as training on concrete surface. Incorrect predictions in Figure 9 also show that currently masonry surface, shadows and narrow cracks are not well excluded and identified by this model. It is necessary to introduce more similar data during the training process to improve the model.

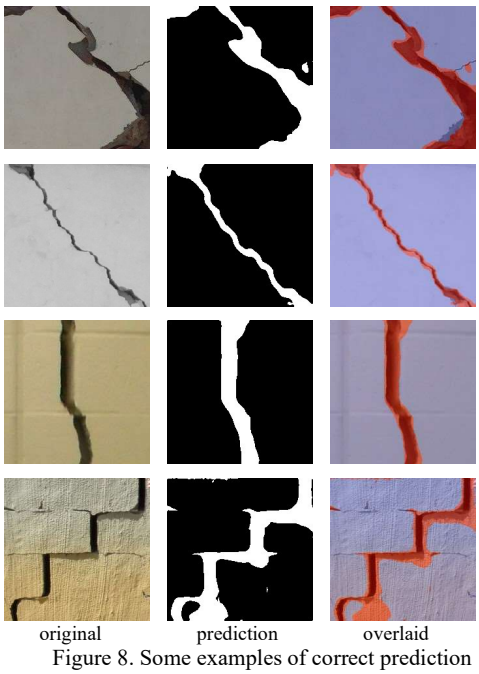

Figure 8. Some examples of correct prediction

4.2 Object-level Data in Phi-Net

At object-level and structural-level scales, cracks become narrow, long and less visible since they are far from the cameras when focal length is fixed. In addition, these images also include some other objects and non-cracking damage, which will make the feature extraction easier to be distracted and misclassified. We established a principle that the selected images are typical cracks with these distractions and can be used as templates. Moreover, we tried to keep a balanced training data, i.e. the equal number of images for two levels are chosen. However, the total number of labeling images are still small and that goal was not achieved in our training.

The accuracy for validation step reaches up to 98.82%. Then we directly tested the object-level training data in Phi-Net, including 5,713 images with cracks and without any crack. The test results show that our model can segment cracks on 1,494 images while it fails on 4,219 images, most of which are non-cracking one but are wrongly labeled as cracked one, the accuracy is only 26.15%.

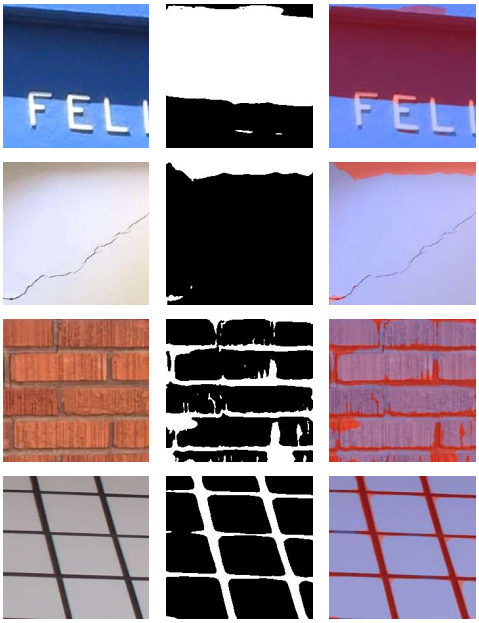

Figure 9. Some examples of incorrect prediction

Examples of good prediction and mis-prediction can be seen in Figures 10 and 11. However, after using our ResNet with Task 8 to filter the non-cracking images, we obtained 1,896 images which are identified as cracked. Then the U-Net was employed to locate the cracks again. 1,129 images can be well predicted the location of the cracks, and the accuracy is improved to 59.55%. This test shows that in this research it was necessary to improve the accuracy of prediction by cascading two networks.

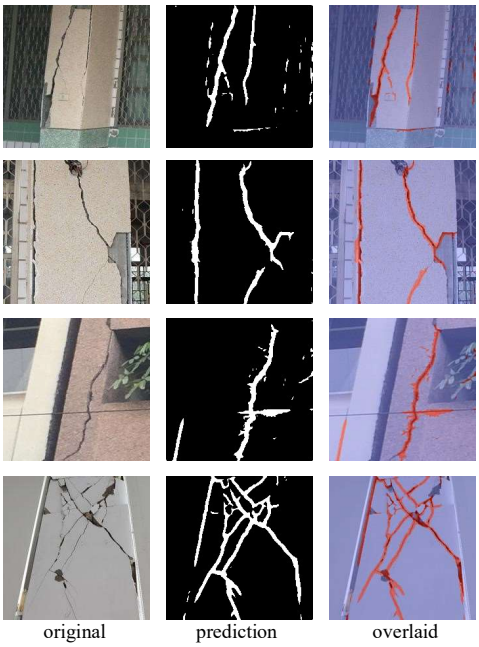

Figure 10. Some examples of good testing results for object-level Phi-Net data

As shown in Figures 10 and 11, cracks on the beams, columns and walls are masked while the scales are also varied within object level. It should be pointed out that there are much more noise in prediction due to distraction by different crack-like objects, such as cables and wires, even some plants (see Figure 11). On the other hand, most of the cracks are 3D ones in the testing set compared to the previous training and testing process, which are more like a mission on planar object detection so that the network just needs to focus on 2D feature learning.

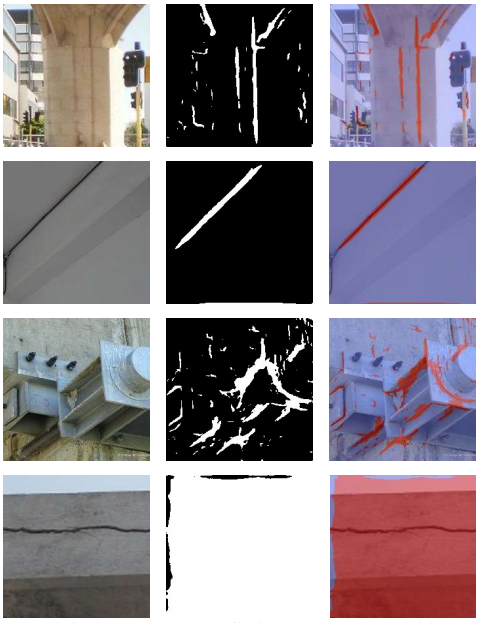

Figure 11. Some examples of incorrect testing results for object-level Phi-Net data

4.3 Structural-level Data in Phi-Net

In a structural-level dataset, 5,832 images in Phi-Net are used to detect cracks in various buildings and bridges. However, the task becomes more complicated with inclusion of people, plants, pavements and other objects. Compared to these objects, cracks are tiny and more likely to be distracted and occluded by crack-like objects like wires, cables and other damage like spalling and exposed reinforcements.

Figure 12. Some examples of good testing results for structural-level Phi-Net data

The test results show that our model can predict cracks in 500 images while it fails on 5,332 images, most of which are non-cracking ones but being mis-predicted as cracked ones, the accuracy is only 8.57%. Examples of fair prediction and mis-prediction can be seen in Figures 12 and 13. Our model identified 717 images as the cracked when our ResNet with Task 8 was employed to gate the non-cracking ones. So the U-Net can predict the location of the cracks in 356 images, thus the model achieves an accuracy of 49.65%. The cascaded networks improve the accuracy at this level as well as at object level.

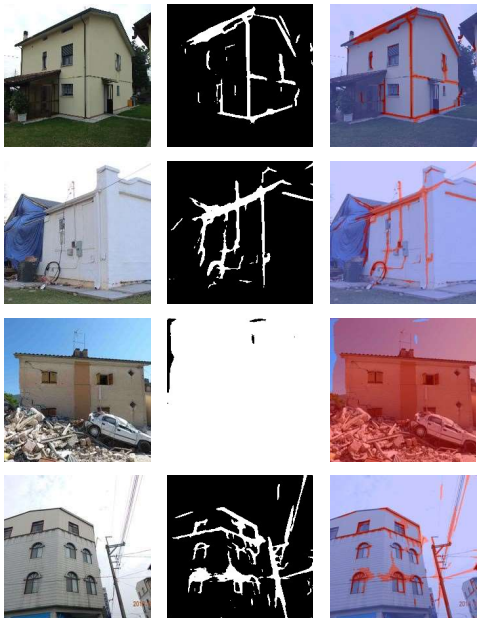

Figure 13. Some examples of incorrect testing results for structural-level Phi-Net data

It should be noted that most of cracks are smoothed after being resized from original and high resolution images into these low resolution images, since the image size is just 224×224 for testing. And it is also an outcome for labeled images are not sufficient in training now. Moreover, angle or viewpoints of cracks are quite different from the training data because hundreds of these labeled images cannot cover them all in various complicated scenes.

It is difficult for our U-Net model to learn geometry of the cracks and texture features from these limited images and to localize the cracks with less noise and errors at this scale now. Therefore, increasing labeled training data will be a way to improve the process in our future work.

5. DISCUSSION

In this research, two deep learning neural networks and two datasets are used to test the proposed method pipeline to classify and segment cracks at various scales, which are defined and separated because their characteristics are so different. It is shown that the use of these cascaded networks on semantic segmentation for cracks and other types of structural damage is possible.

1) At pixel level, cracks are not fully shown but occupy a big portion of images, and their discontinuity in width and continuity in length are distinctive. Therefore, a very high accuracy is achieved with less errors and noise to detect them by the U-Net.

2) For object-level cracks, the cracks appear long and narrow, and also show the discontinuity on the material surface of structural or nonstructural objects. Furthermore, various common objects are captured in this scene and angle of views on the cracks also changes between different images. Some of cracks in this scale may not be planar-like as at pixel level. This brings challenges to precisely locate the cracks on various materials. Although there are some noise in predictions, which are caused by insufficient training data to cover all the similar scenarios in testing data, especially for some steel and masonry structures, the results show our model can locate the cracks with a higher accuracy with the cascaded networks.

3) At large scale, defined as structural level in this study, compared to entire buildings and bridges, cracks are more likely to be invisible. Some of them are wide and long enough to be detected with vision-based technologies. However, more non-related objects are common in images, and some of them are very similar to cracks. Furthermore, there are more 3D cracks instead of planar ones, with not enough data for training. On the other hand, with lower resolution, there are more noise and incorrect predictions on the cracking localization. But the proposed method also increases the accuracy dramatically at this scale.

4) The proposed method has the potential to semantically segment the cracks and other structural damage in the future. For example, if Task 1, 6 and 8 in the ResNet are employed to classify the cracking and implement U-Net to locate the cracks on a column successfully, then we can label it like this: "*This is a column with shear-damaged cracks, and cracks are shown in red markers in the image*".

6. CONCLUSIONS

In this paper, two kinds of neural networks are proposed for structural damage detection and cracking localization. Most CNNs like our ResNet model for SDD cannot identify the locations of structural damage. A solution is provided for this problem by introducing another network, the U-Net, to locate the damage. However, currently we just have to try to locate the cracks at various scales. In our methodology, the ResNet is used to classify the scene levels and gate the noncracking and cracking at first, then the U-Net is employed to locate these cracks. In our experiments, after training 1,000 images out of a dataset with 20,000 images, the proposed method can give a very high accuracy to mask the cracks on pixel-level images based on two open-source image datasets. We believe this is because the scene is simple and most of the cracks are planar.

We labeled 853 object-level and structural-level images and train with the U-Net model, then test the data from Phi-Net. Although it can give fairly good prediction to detect the location of cracks, the errors and noises increase due to presence of more objects as distraction and 3D spatial effects under such scales. Moreover, we found out that the strategy to use the U-Net model as an end-to-end network to classify and locate the cracks under these large scales doesn't work, but the accuracy has been improved significantly when the ResNet is used to pick up those images with cracks first and then mask them by the U-Net. Therefore, the proposed method is a right solution for segmentation problem on detecting structural damage, especially with limited training data now.

Our training datasets are available from this link:
https://github.com/OSUPCVLab/StructureCrackDataset.

REFERENCES

Ali, L., Khan, W., Chaiyasarn, K., 2019. Damage Detection and Localization in Masonry Structure Using Faster Region Convolutional Networks. *International Journal OF GEOMATE*, 17(59), 98–105.

Atha, D. J., Jahanshahi, M. R., 2018. Evaluation of deep learning approaches based on convolutional neural networks for corrosion detection. *Structural Health Monitoring*, 17(5), 1110–1128.

Brooks, J., 2019. COCO Annotator. https://github.com/jsbroks/coco-annotator/.

Cha, Y.-J., Choi, W., Büyüköztürk, O., 2017. Deep learning- based crack damage detection using convolutional neural net- works. *Computer-Aided Civil and Infrastructure Engineering*, 32(5), 361–378.

Dung, C. V., L. D., Anh, 2019. Autonomous concrete crack detection using deep fully convolutional neural network. *Automation in Construction*, 99, 52–58.

Flôr, A., 2019. arthurflor23/surface-crack-detection. https://github.com/arthurflor23/surface-crack-detection.

Gao, Y., Mosalam, K. M., 2018. Deep transfer learning for image-based structural damage recognition. *Computer-Aided Civil and Infrastructure Engineering*, 33(9), 748–768.

Gao, Y., Mosalam, K. M., 2020. New peer report 2019/07:" peer hub imagenet (Ø-net): A large-scale multi-attribute benchmark dataset of structural images". https://peer.berkeley.edu/news/new-peer-report-201907-peer-hub-imagenet-Ø-net-large-scale-multi-attribute-benchmark-dataset.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learn- ing for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

Hoskere, V., Narazaki, Y., Hoang, T., Spencer Jr, B., 2017. Vision-based Structural Inspection using Multiscale Deep Con- volutional Neural Networks. In *Proceedings of 3rd Huixian Int. Forum on Earthquake Engineering for Young Researchers*. Urbana, IL: Univ. of Illinois at Urbana Champaign.

Kong, X., Li, J., 2018. Automated fatigue crack identification through motion tracking in a video stream. *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2018*, 10598, International Society for Optics and Photonics, 105980V.

Liu, Y., Yao, J., Lu, X., Xie, R., Li, L., 2019a. DeepCrack: A deep hierarchical feature learning architecture for crack seg- mentation. *Neurocomputing*, 338, 139–153.

Liu, Z., Cao, Y., Wang, Y., Wang, W., 2019b. Computer vision-based concrete crack detection using U-net fully convolutional networks. *Automation in Construction*, 104, 129–139.

Özgenel, C. F., 2018. Concrete Crack Images for Classification. *Mendeley Data, v1 http://dx. doi. org/10.17632/5y9wdsg2zt*, 1.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. *International Conference on Medical image computing and computer-assisted intervention*, Springer, 234–241.

Spencer Jr, B. F., Hoskere, V., Narazaki, Y., 2019. Advances in computer vision-based civil infrastructure inspection and monitoring. *Engineering.* 5 (2): 199–222

Yang, L., Li, B., Li, W., Liu, Z., Yang, G., Xiao, J. 2017. Deep concrete inspection using unmanned aerial vehicle towards cssc database. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 24-28.

Yeum, C. M., Dyke, S. J., Ramirez, J., 2018. Visual data classify-cation in post-event building reconnaissance. *Engineering Structures*, 155, 16–24.

Zha, B., Bai Y., Sezen, H., Yilmaz, A., 2019. Deep Convolutional Neural Networks for Comprehensive Structural Health. *Structural Health Monitoring 2019*, 3367-3374.

Zhang, A., Wang, K. C., Fei, Y., Liu, Y., Tao, S., Chen, C., Li, J. Q., Li, B., 2018. Deep learning–based fully automated pavement crack detection on 3D asphalt surfaces with an improved CrackNet. *Journal of Computing in Civil Engineering*, 32(5), 04018041.