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

SEMANTIC SEGMENTATION FOR VISIBLE SEISMIC DAMAGE OF REINFORCED CONCRETE STRUCTURES

Z. Miao[(1)], X. Ji[(2)], Y. Zhuang[(3)]
[(1)] *Graduate student, Department of Civil Engineering, Tsinghua University, Beijing, China, miaozh17@mails.tsinghua.edu.cn*
[(2)] *Associate professor, Department of Civil Engineering, Tsinghua University, Beijing, China, jixd@mail.tsinghua.edu.cn*
[(3)] *Graduate student, Department of Civil Engineering, Tsinghua University, Beijing, China, zhuangyc19@mails.tsinghua.edu.cn*

Abstract

Visible seismic damage of reinforced concrete (RC) structures, including crack, spalling and crushing of concrete, exposure and buckling of reinforcement, provides profound insights into understanding damage mechanism and evaluating damage states of RC structural components. Thus, the damage identification (i.e., detection, classification and quantification of various types of visible damage) plays an important role in the post-earthquake assessment of RC building structures. Current practice for post-earthquake damage identification relies on eye-inspection, which remains subjective and time-consuming.

In this study, a novel approach for identification of visible damage of RC structures is proposed, based on computer vision theories and algorithms (e.g., semantic segmentation). First, a semantic segmentation database was constructed from test photos of RC structural specimens, including shear walls and joints. The generality and representativity of the database were examined. Second, based on the architectures of the well-known convolutional networks, i.e., VGG-16 and U-Net, a deep convolutional network for semantic segmentation of visible damage was designed, named as Damage-Net. Two sets of parameters for the proposed Damage-Net were optimized separately for the detection of cracks and other types of damage, named as Crack-Net and 4Cate-Net, respectively. The transfer learning technique was utilized in the training process of convolutional networks, enabling reduced computing resources and database size. Third, a post-processing approach for the outputs of Crack-Net, i.e., Threshold after Histogram Equalization, TaHE, was developed to improve the performance of crack identification. Comparison of crack characterization results suggested that, the proposed TaHE was capable of providing improved results for crack width quantification. In the end, the proposed vision-based approach was applied to a series of photos of a beam-to-wall joint specimen. It is indicated that damage metrics (e.g., crack width, spalling size, exposure of reinforcement) can be easily obtained from the segmentation results of visible damage.

In summary, the proposed vision-based approach for RC structural damage identification achieves competitive results with respect to accuracy and generalization. The approach can provide fundamental support for further development of an autonomous, robust and efficient post-earthquake assessment program for RC building structures.

Keywords: visible damage identification; computer-vision; semantic segmentation; convolutional neural networks; RC structures.

1. Introduction

Post-earthquake safety assessment of building structures has played a critical role in emergency treatment and after-hazard restoration of urban communities. The current practice of safety assessment relies on field surveys carried out by certified structural engineers or professionals, which causes the procedure extensively time-consuming. Moreover, the existing procedure for the safety assessment of building structures is severely subjective, since the evaluation results are primarily based on the professional knowledge and experience of different inspectors.

Meanwhile, significant progress has been achieved in the fields of computer vision and machine learning in recent years. By taking advantage of the algorithms of these fields, there is a high potential for the development of an autonomous post-earthquake safety assessment system for building structures, which would improve current practice for both efficiency and accuracy.

According to the guidelines for post-earthquake safety assessment in China [1], Japan [2, 3] and US [4], visible damage of RC structural components is closely related to the degradation of their mechanic properties, and is used as the essential information for the damage state estimation of RC components. For this reason, visible damage identification should be considered as a fundamental step in the overall safety assessment procedure for building structures.

In this research, techniques from computer vison, e.g., semantic segmentation using convolutional networks, are leveraged to solve the puzzle of visible damage identification for RC structural components. The objectives of this research are (1) to localize, classify and segment typical damage of RC components, (2) to achieve accurate quantitative characterization of cracks, including crack width, angle and length.

2. Fundamentals about convolutional networks

First of all, to effectively organize the contents of this paper, and to facilitate the understanding, a brief introduction about convolutional networks is provided. A convolutional network can be considered as a special type of data-fitting function (or model). The mathematical formulation of a convolutional network is determined by its architecture, where the organization of series of basic operations, including 2D-convolution, ReLU and max-pooling, is defined. The convolutional network takes an image of size $H \times W$ as input, and for the task of segmentation, it outputs a matrix of size $H \times W \times K$, which consists of the predictions for the $H \times W$ pixels. The prediction for each pixel, which is a K-dimensional vector, i.e., $\boldsymbol{p}_{ij} = [p_{1,ij}, p_{2,ij}, \dots, p_{K,ij}]$, is interpreted as a discrete probabilistic distribution, where $p_{k,ij}$ represents the predicted probability that the pixel at coordinate (i, j) is an instance of category k.

Data with annotated ground truth, referred to as a dataset, is required to assign reasonable values for the parameters of the model. The subset of the dataset that is used for the optimization (i.e., the training) of the parameters is called the training set, and the rest, which is used to validate the performance of the trained model, is called the test set. The training process of the model is to solve an optimization problem by the stochastic gradient descent (SGD) algorithm [5, 6] or one of its variants, such as Adam [7] and RMSProp [8]. For a model applied for semantic segmentation, the objective function at iteration t (i.e., the loss of the model) is demonstrated in Eq. (1).

$$L_t = \frac{1}{N} \sum_{s=0}^{N-1} l(\boldsymbol{p}_s^{(t)}, c_s) \tag{1}$$

In Eq. (1), N denotes the number of pixels in the training set, and s is an iterator among the pixels. The term $l(\boldsymbol{p}_s, c_s)$ is referred to as the loss function, which evaluates the error between the prediction at iteration t and the ground truth category of the sth pixel. For the SGD and its variants, the objective function is approximated by the samples in a small subset of the training set (i.e., a batch) at each iteration, as denoted in Eq. (2), where N_b denotes the number of pixels in a batch. A batch of samples is randomly selected from the

training set at each iteration, and the number of samples in a batch (i.e., the batch size) is pre-assigned hyper-parameter of the training. A small batch size can lead to a noisy approximation of the objective function and the corresponding gradient, which further causes the training slow-convergence or even divergence.

$$\tilde{L}_t = \frac{1}{N_\mathrm{b}} \sum_{s=0}^{N_\mathrm{b}-1} l(\boldsymbol{p}_s^{(t)}, c_s) \tag{2}$$

3. Database for visible damage of RC components

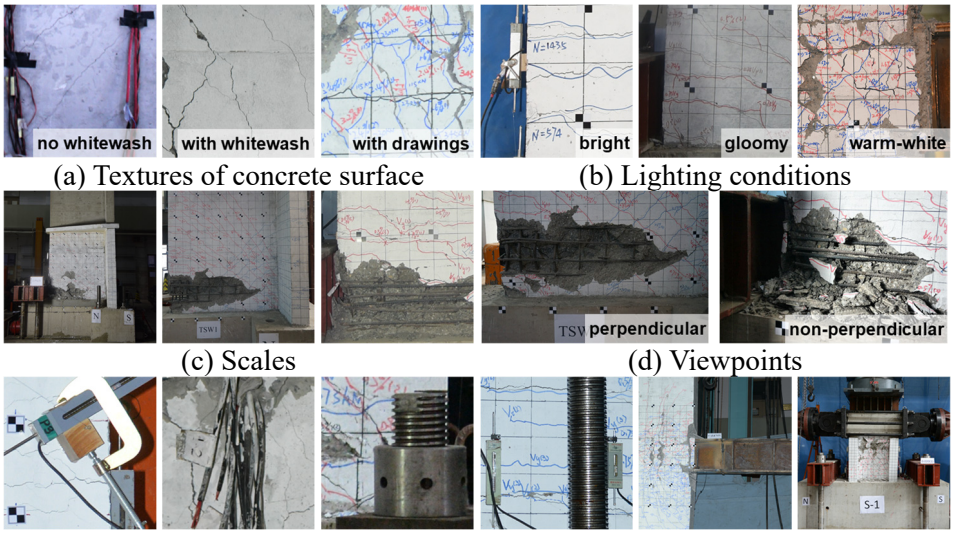
Data is playing a fundamental role in the achievements of deep learning. In this research, a semantic segmentation database that has wide diversity in terms of appearances, scales, viewpoints, background clutter and occlusions, is collected and constructed for visible damage of RC components.

3.1 Database construction

The construction of database is divided into three key steps: (1) Selection of target categories; (2) Sample collection for the representation of selected categories; (3) Manual annotation of collected images. Special issues and considerations for each step are discussed as follows.

For the selection of target categories, types of damage that are thought to be, with general agreement, most typical and effective for the estimation of damage state are focused on, since the purpose of visible damage identification is to provide reliable evidence for the damage state estimation of RC components. As a result, the set of target categories includes the surface **Crack**, **Spalling** and **Crushing** of the concrete, **Exposure** and **Buckling** of the reinforcement.

Image Samples are selected from high-resolution test photos of a broad range of RC test specimens including RC shear walls and beam-to-wall joints. Diversity of the collected samples is ensured and promoted through the following ways. First, the selected specimens were of different section shapes, shear-to-span ratios, reinforcement configurations and expected failure modes. Second, various cameras with different configurations were used in the tests, and thus diversity is enlarged in terms of scales and viewpoints. Third, sample selection is conducted so that various background clutter and occlusions would be included into the database. Several samples are demonstrated in Fig. 1 to visualize the diversity of the constructed database from multiple perspectives.



(a) Textures of concrete surface (b) Lighting conditions

(c) Scales (d) Viewpoints

(e) Background clutter and occlusions

Fig. 1 - Diversity of the constructed database.

For the annotation of the image samples, a guideline is summarized based on the observation of visible damage and the understanding of the damage mechanisms of RC structural components, in order to ensure the accuracy and consistency of the manual annotation. Manual annotation of the collected samples is carried out following the guideline. Strict and detailed check on the annotated results is performed by graduates majored in civil engineering. During the process of annotation and check, ambiguous cases are discussed and confirmed with professors and experienced engineers. Accuracy is emphasized in the entire workflow of annotation, through which the high quality of the constructed database is guaranteed.

3.2 From database to datasets

Image samples in the constructed database are of arbitrary sizes, i.e., the heights and widths of the images are different from each other. Rather than feeding the samples from the database directly to the convolutional networks, two aspects suggest the usage of samples with uniform size. First, the training of convolutional networks for semantic segmentation is highly memory expensive. By reducing the sample size to a relatively small and uniform size, a larger batch size can be configured, and thus, a faster and better convergence of the training model can be achieved. Second, the number of image samples contained in the constructed database is limited, meanwhile, the sample size of each image is rather large. Effective data-balancing, which is a crucial factor in the training of the convolutional networks, is therefore restrained. Issues of data imbalance and strategies for data-balancing will be discussed in detail later in this subsection.

To overcome the aforementioned issues, datasets are constructed from the database. In the context of this research, the term “database” is used to denote a set of image samples which occupy arbitrary sample sizes, and the term “dataset” is used to denote a set of image samples with a standard and uniform size (i.e., 300 pixel×300 pixel in this research). Sample candidates for the datasets are generated following the procedure demonstrated in Fig. 2, where techniques for data augmentation is introduced in the procedure to exploit data diversity. By randomly selection of center point, rotation angle, scaling factor and flipping axis, the diversity of the datasets is enriched in terms of certain affine transformations, including translation, rotation, scaling and reflection.

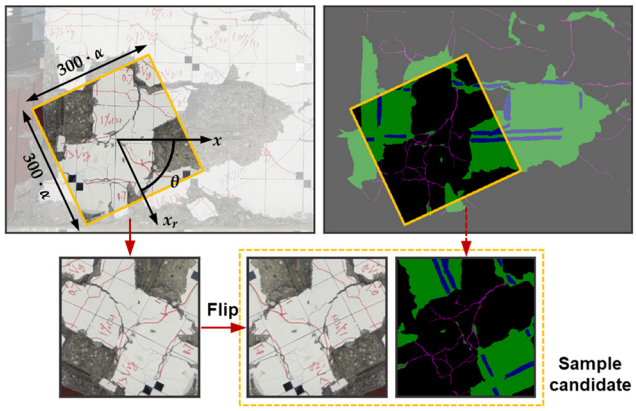


Fig. 2 - A flowchart for the generation of the sample candidates.

Data imbalance is an intrinsic issue for the collected database. At the sample level, the number of image samples which contain concrete crushing and reinforcement buckling areas is limited, since severe damage occurs at the final stage of the loading procedure. Moreover, the characteristics of damage categories in spatial distribution lead to severe data imbalance at the pixel level. In addition to the fact that the pixels of background are dominant in the database, the pixels of concrete damage (i.e., spalling and crushing) tend to occupy an area, while the pixels of reinforcement damage (i.e., exposure and buckling) are only scattered in this area. Especially, the crack category requires additional attention, as it occupies a proportion of pixels of merely 3% in the database.

Data imbalance can extensively affect the training process. If the convolutional networks are trained using the data which suffers from severe data imbalance, the resulting model would show drastically distinct

recognition for different target categories. Several data-balancing strategies are applied to the generation of the datasets, which are able to mitigate the impact of data imbalance at various aspects and levels.

(1) Individual dataset for crack detection, the Crack-DS

Compared with other four categories, the pixel-level proportion of cracks is almost negligible, and the visual characteristics of crack category appear to be extremely local and detailed. An individual dataset, named as the Crack-DS, is therefore generated for crack detection. The dataset that is used for the detection of the other four categories is referred to as the 4Cate-DS.

(2) Sample-level data-balancing

Data imbalance at the sample level is mitigated by conducting category-existence examination on sample candidates before including them into the datasets. For the crack category, a sample candidate is accepted as a sample of the Crack-DS only if cracks are contained in this candidate.

For the other four categories, samples are firstly generated into two separate groups, denoted as SpEx-Group (Spalling/Exposure Group) and CrBk-Group (Crushing/Buckling Group), respectively. As there are more images containing spalling and exposure category in the original database, in order to ensure the number of samples in the two groups are comparable with each other, a lower sampling rate is assigned for the generation of the SpEx-Group. The 4Cate-DS is eventually constructed as the combination of the SpEx-Group and the CrBk-Group.

For both the Crack-DS and the 4Cate-DS, 85% samples are randomly picked to compose the training set, and the rest of the samples compose the test set. Statistical information of the resulted datasets is listed in Table 1.

Table 1 - Statistical information of the datasets.

Dataset	**Category**	**Number of samples** (Training / Test set)	**Proportion of pixels** (Training set)
4Cate-DS	Background	1046 / 184	55.70%
	Spalling	740 / 128	17.54%
	Exposure	559 / 98	2.12%
	Crushing	531 / 90	20.69%
	Buckling	471 / 85	3.95%
	Total	1046 / 184	-
Crack-DS	Background	2018 / 356	98.22%
	Crack		1.78%

(3) Pixel-level data-balancing

The strategy proposed by Eigen and Fergus [9] is applied to resolve the pixel-level data imbalance, where the weighting coefficients are introduced to the objective function. The weighted objective function is formulated in Eq. (3). The weighting coefficient for each category is calculated as in Eq. (4),

$$\tilde{L}_t = \frac{1}{N_\mathrm{b}} \sum_{s=0}^{N_\mathrm{b}-1} w_{c_s} \cdot l(\boldsymbol{p}_s^{(t)}, c_s) \tag{3}$$

$$w_k = \frac{1}{K} \cdot \frac{N}{N_k} \tag{4}$$

where K denotes the number of categories, N is the number of pixels in the training set, and N_k is the number of pixels of category k (in the training set). Compared with the original objective function, the weighted

objective function emphasizes the importance of the small-portion categories, and thus guides the trained model into comparable recognition among the target categories.

4. Damage identification model

4.1 Damage-Net

In general, the architecture of convolutional networks used for semantic segmentation is composed of the encoder, the decoder and a series of skip connections, as in Fig. 3 illustrated. The encoder functions exactly as the convolutional layers in the models for classification, and its structure is often inherited from classification convolutional networks. In the task of segmentation, the encoder is responsible for coarsely predicting the categories of each region in the input image. The size of the prediction obtained from the encoder is enlarged by the decoder to the size of the input. During the zooming process, multiscale local information is supplied through skip connections to assist the reconstruction of a finer prediction with respect to the local boundaries between areas of different categories.

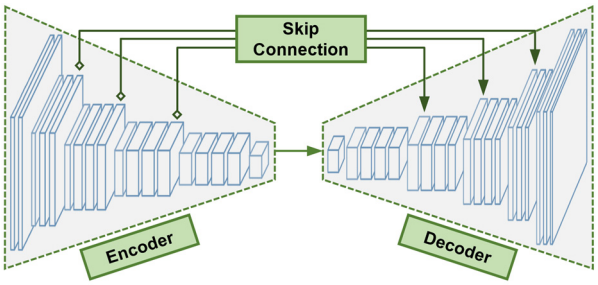


Fig. 3 - The general architecture of convolutional networks for semantic segmentation.

Among various state-of-art models for segmentation, such as FCN [10], U-Net [11], DeepLab [12] and SegNet [13], the major differences lie in the design of the decoders and skip connections. The architectures of these models are examined and compared during the pre-test, where it is noticed that the U-Net suits the purpose of visible damage detection best, since it provides a direct and intact way for local information integration.

An architecture of a deep convolutional network, named as Damage-Net, is designed specifically for visible damage detection of RC components. The architecture is based on U-Net, while adaptive improvements are carried out in terms of flexibility and training efficiency.

First, the output prediction in U-Net is not of the same size as the input image, which makes the data pre- and post-processing a bit cumbersome. In the proposed Damage-Net, layer configurations, such as padding size and stride size, are precisely decided, in order to ensure that the input and the output are of the same size. Second, the encoder of U-Net is not inherited from any known architecture of classification models, therefore the parameters need to be optimized from scratch. Training from scratch requires more training data, and may be confronted with issues like over-fitting and slow-convergence. Damage-Net, by contrast, learned from the design of FCN, SegNet and DeepLab, inherits its encoder from the convolutional layers of VGG-16 [14], a deep convolutional network which achieved excellent accuracy on the large-scale, general-purpose dataset ImageNet [15]. The adaptation from VGG-16 enables Damage-Net to conduct transfer learning [16, 17], which leads the training on relatively small-scale datasets to a better convergence with less computing resources.

The architecture of Damage-Net is illustrated in Fig. 4. Four skip connections are introduced to effectively integrate multiscale local information for delicate boundary construction. Damage-Net has 28.8 million parameters in total, and 14.7 million are deducted if the parameters are transferred from VGG-16.

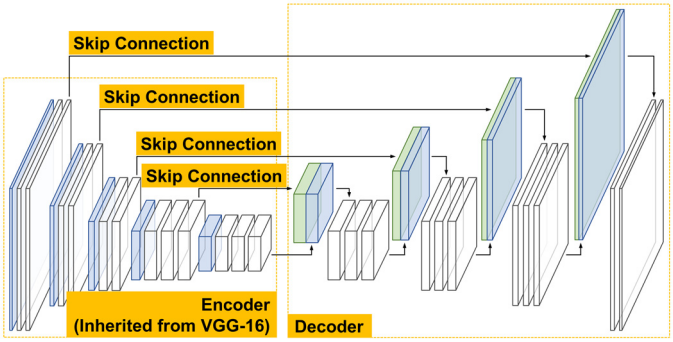


Fig. 4 - The architecture of the proposed Damage-Net.

4.2 Training and performance evaluation of Damage-Net

As mentioned in Section 2, crack category is separated from the other four categories to form its individual dataset Crack-DS. Therefore, two types of models, denoted as 4Cate-Net and Crack-Net separately, are optimized correspondingly.

Models with different training configurations are compared to demonstrate the superiority of the proposed architecture Damage-Net. The models will be evaluated on aspects of resource consumption (i.e., computing time and memory usage) and recognition performance. The performance of a model is reported on the test set using standard and well-known metrics including pixel accuracy (PA), per-category pixel accuracy (perPA), mean pixel accuracy (MPA), per-category intersection over union (perIoU), mean intersection over union (MIoU) and frequency-weighted intersection over union (FWIoU). The detailed formulation and explanation of these metrics can be found in the reference [18]. Among them, the MIoU is considered as the main indicator for accuracy evaluation in this research, since it is the most used and accepted metric due to its representativeness and simplicity.

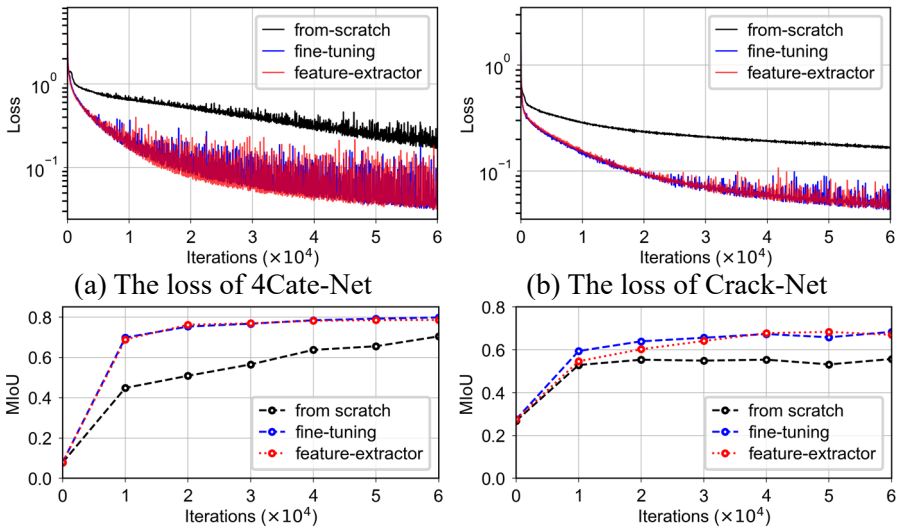
Transfer learning is a practical technique for training convolutional networks on relatively small-scale datasets with limited computing resources. In brief, transfer learning is the application of certain parameters from a pretrained model into the target model, where part of the architecture is shared between the two models. The reason for parameter “transferring” among models is that the recognition of visual features such as basic geometric shapes and simple textures may form the cornerstone of any visual systems, although the target domains of the systems vary significantly. Most commonly, the pretrained model is one that trained on large-scale datasets (e.g., OpenImage [19], ImageNet [15]), and thus is considered to preserve broad generalization and possess excellent recognition of basic visual features.

Two strategies of transfer learning can be deployed while training a convolutional network. The first one is referred to as the fine-tuning strategy [17], where parameters transferred from the pretrained model are updated iteratively using a relatively small learning rate. The second one is called the feature-extractor strategy [16], where transferred parameters are fixed (i.e., frozen), and only the newly configured and randomly initialized parameters of the target model are optimized during the training.

In order to show how the training process is benefited from the architecture of Damage-Net, numerical tests are carried out to analyze the differences in performance, computing time and memory usage among the networks that trained without transfer learning (i.e., from scratch) and with two different transfer learning strategies. The models are trained under the same configuration using the RMSProp algorithm, while the only difference is how the parameters are initialized and updated. As for the from-scratch model, all the parameters are randomly initialized and iteratively optimized during training. For the two transfer-learning models, transferred parameters are initialized from the corresponding VGG-16 layers, while the rest are randomly initialized. During the training, transferred parameters in the fine-tuning model are updated with a smaller learning rate (i.e., 1% of the base learning rate), and the ones in the feature-extractor model are fixed.

Performance comparison among the three models are shown in Fig. 5. Training from scratch is less effective than applying transfer learning concerning the accuracy of the resulted models. Between the two

strategies, the corresponding models have shown considerably similar detection accuracy, while the feature-extractor strategy is more computationally economic, since it cut down the training time by about 30% and spare the memory usage by about 25%, as demonstrated in Table 2.



(a) The loss of 4Cate-Net (b) The loss of Crack-Net

(c) MIoU of 4Cate-Net (d) MIoU of Crack-Net

Fig. 5 - Performance comparison of models trained from scratch, with fine-tuning strategy and with feature-extractor strategy.

Table 2 - Computing resources required by models trained from scratch, with fine-tuning strategy and with feature-extractor strategy.

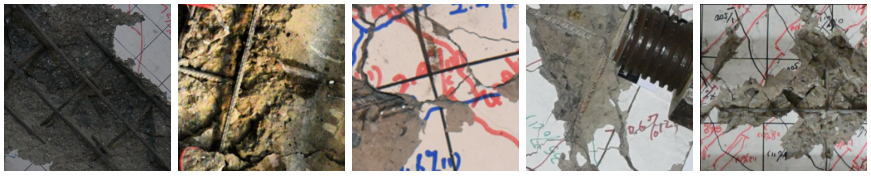
Network	**Strategy**	**Computing time per iteration** / ms (mean $\pm 1 \cdot \sigma$)	**Memory usage per sample** / MB
4Cate-Net	From scratch	1647±56	820
	Fine-tuning	1760±81	820
	Feature-extractor	1227±35	626
Crack-Net	From scratch	1600±47	813
	Fine-tuning	1707±77	813
	Feature-extractor	1220±38	619

Eventually, a few models are trained using the Adam and RMSProp algorithms separately, and models with the highest MIoU are selected as the ultimate sets of parameters for 4Cate-Net and Crack-Net, which will be used for subsequent analysis in this article. Metrics of the ultimate models on the test set are reported in Table 3.

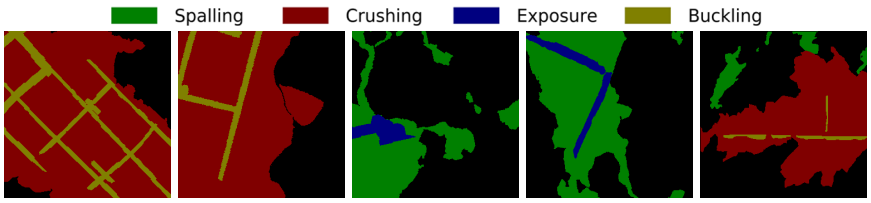
Table 3 - Performance of the ultimate models.

Network	**Category**	**PA** /%	**perPA** /%	**FWIoU** /%	**perIoU** /%
4Cate-Net	Background	92.42	95.58	86.21	91.92
	Spalling		88.10		79.30
	Exposure		76.04		60.77
	Crushing		91.24		82.69
	Buckling		84.06		71.89
	Average	**MPA** /%	87.00	**MIoU** /%	77.31
Crack-Net	Background	97.98	98.42	96.85	97.95
	Crack		76.84		43.35
	Average	**MPA** /%	87.63	**MIoU** /%	70.65

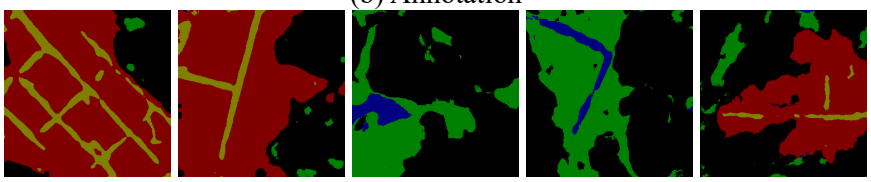
Several samples from the test set are demonstrated in Fig. 6 and Fig. 7 to visualize the performance of the 4Cate-Net and Crack-Net, respectively.



(a) Image samples

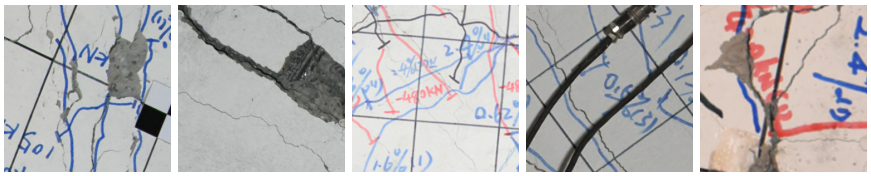


(b) Annotation

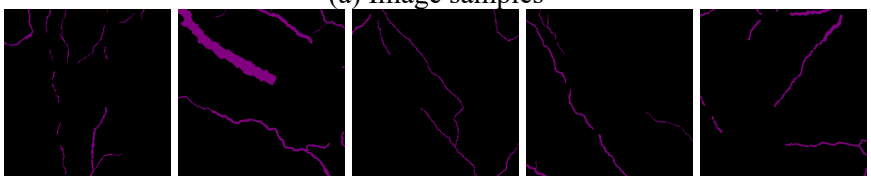


(c) Output of 4Cate-Net

Fig. 6 - Performance visualization of 4Cate-Net.



(a) Image samples



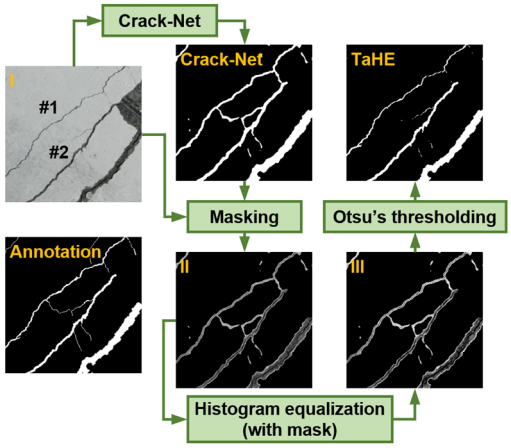
(b) Annotation

(c) Output of Crack-Net

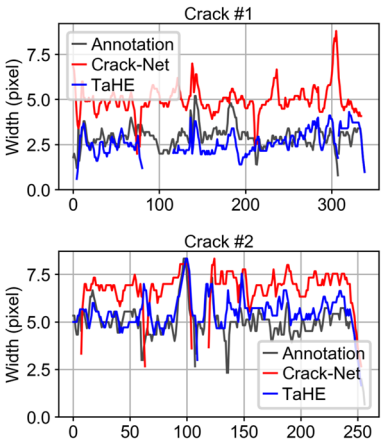
Fig. 7 - Performance visualization of Crack-Net.

4.3 Post-processing

Although the Crack-Net is remarkable for accurately detecting cracks in the sense of localizing the cracks and tracking the paths of the cracks, it has been noticed that the convolutional network has difficulty in precisely delineating the boundaries of cracks. Instead, the model tends to extract a crack together with borders surrounding it, and thus causes the overestimation of crack width. A post-processing technique is proposed in this study to cope with this defect, as in Fig. 8(a) illustrated.



(a) Flowchart of the proposed Threshold after Histogram Equalization



(b) Comparison of crack width analyzed from Annotation, Crack-Net and TaHE

Fig. 8 - The proposed post-processing technique, TaHE, and its performance.

The basic idea is to use the output of Crack-Net as a mask for crack detection, and the boundaries of cracks can further be determined through the contrast of intensities between crack-pixels and background-pixels. Specifically, the original image and the output of Crack-Net is overlapped to get the masked image (see Image II in Fig. 8(a)), and then the Histogram Equalization with Mask algorithm [20] is conducted to improve the global contrast of the masked image. As follows, the Otsu's thresholding [21] is applied to the equalized masked image (see Image III in Fig. 8(a)), which leads to the detected cracks with refined boundaries. The proposed technique is named as Threshold after Histogram Equalization (TaHE) according to the procedure of the technique.

As shown in Fig. 8(b), cracks depicted in the images of (manual) annotation, (the output of) Crack-Net and (the result of) TaHE are characterized using algorithms proposed in the study [22] to validate the performance of the proposed TaHE. The quantitative results of crack width obtained from the annotation and TaHE are close to each other, while the output of the Crack-Net overestimates the crack width by a significant margin. The analysis results suggest the effectiveness of the proposed TaHE.

5. Application on structural specimen photos

The proposed vision-based damage identification models are applied to a series of test photos of a beam-to-wall joint specimen (see Fig. 9(a)). Perspective transformation [23] is utilized as pre-processing to correct the lens distortion of the original photos, and the conversion factor between pixel-unit and engineering-unit is further derived from the corrected photos.

The photos are then analyzed by the 4Cate-Net, Crack-Net and the corresponding post-processing technique TaHE, and the results are visualized in Fig. 9(b). The size of the spalling region is directly calculated by accumulating spalling pixels detected in each test photo, and the development of spalling area along the loading process are plotted in Fig. 9(c).

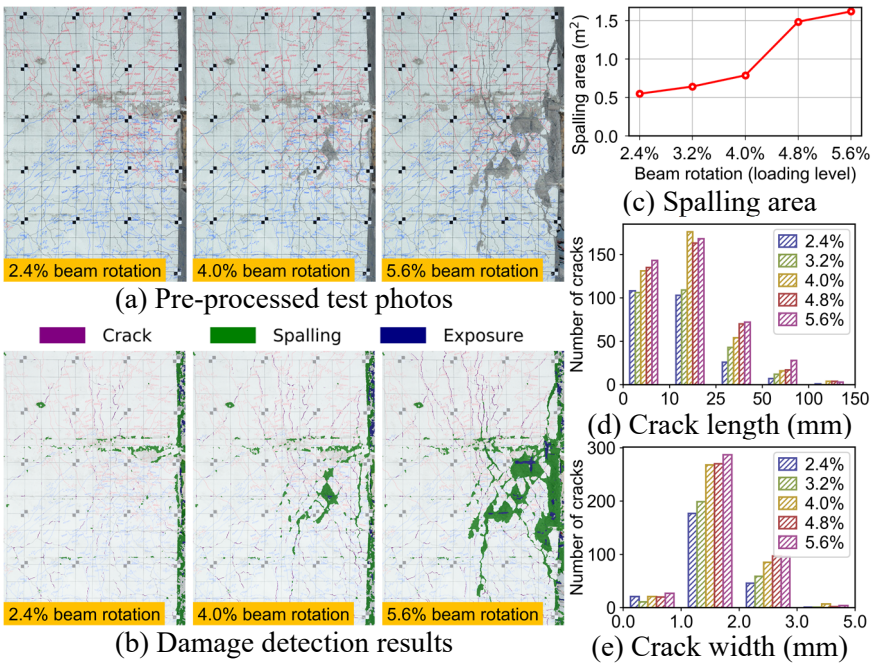


Fig. 9 - Application to test photos of a RC joint.

Crack fields identified by Crack-Net and TaHE are further separated, labeled and characterized by the crack characterization algorithm [22], through which, essential geometric properties (i.e., crack length, crack width and crack angle) of each crack are calculated. Statistical distributions of crack length and crack width of the crack field in each loading level are computed and compared in Fig. 9(d) and (e), respectively. It can be observed that the cracks gradually grow longer and wider along the loading levels, which reflects the degradation of the specimen.

6. Conclusions

In this research, a novel vision-based approach is developed for damage identification of RC structural components. Deep learning techniques are deployed in this approach, which enables pixel-level detection for visible damage, including crack, spalling and crushing of concrete, exposure and buckling of reinforcement. The proposed approach is applied to a series of test photos of a beam-to-wall joint specimen, through which, the accuracy and effectiveness are validated.

To achieve semantic segmentation using deep learning techniques, a diverse database is constructed for visible damage of RC structural components. Datasets are further generated where issues of data imbalance are dealt with and data diversity is enriched in terms of certain affine transformations.

Based on VGG-16 and U-Net, a deep convolutional network architecture, the Damage-Net, is proposed, which enables effective training with small-scale dataset. Two models, the 4Cate-Net and Crack-Net, are optimized separately for the detection of crack and other damage. Both models achieve excellent performance with PA above 90% (92.4% for 4Cate-Net, 98.0% for Crack-Net) and MIoU above 70% (77.3% for 4Cate-Net, 70.7% for Crack-Net). Moreover, a simple but effective post-processing technique is proposed for Crack-Net, through which crack width is well-preserved and accurate geometric properties can be further estimated.

7. Acknowledgements

The work presented in this paper was supported by the funds from the National Key R&D Program of China (Grant No. 2017YFC1500602) and National Natural Science Foundation of China (Grant No. 51678347). The authors wish to express their sincere gratitude to the sponsors.

8. References

[1] CMC (2016): *Technical Guide for Post-Earthquake Emergency Assessment of Building Structures (2016 revision)*. China Ministry of Construction. [in Chinese]
[2] JBDPA (1997): *Emergency Risk Assessment Manual for Post-Earthquake Building Structures*. Japan Building Disaster Prevention Association. [in Japanese]
[3] MLIT (2015): *Technical Guide for Damage Estimation and Restoration of Post-Earthquake Buidling Structures (2015 revision)*. Japan Building Disaster Prevention Association. [in Japanese]
[4] FEMA (1998): *Evaluation of Earthquake Damaged Concrete and Masonry Wall Buildings: Basic Procedures Manual*. Federal Emergency Management Agency.
[5] Robbins H, Monro S (1951): A stochastic approximation method. *The Annals of Mathematical Statistics,* 22(3), 400-407.
[6] Zhang T (2004): Solving large scale linear prediction problems using stochastic gradient descent algorithms. *21th International Conference on Machine Learning*, Banff, Canada.
[7] Kingma DP, Ba J (2015): Adam: A method for stochastic optimization. *3rd International Conference for Learning Representations*, San Diego, USA.
[8] Tieleman T, Hinton G (2012): Lecture 6.5-RMSProp: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*.
[9] Eigen D, Fergus R (2015): Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. *2015 IEEE International Conference on Computer Vision*, Las Condes, Chile.
[10] Long J, Shelhamer E, Darrell T (2015): Fully convolutional networks for semantic segmentation. *2015 IEEE Conference on Computer Vision and Pattern Recognition*, Boston, USA.
[11] Ronneberger O, Fischer P, Brox T (2015): U-net: Convolutional networks for biomedical image segmentation. *18th International Conference on Medical Image Computing and Computer Assisted Intervention*, Munich, Germany.
[12] Chen L-C, Papandreou G, Kokkinos I, Murphy K, Yuille AL (2017): Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 40(4), 834-848.
[13] Badrinarayanan V, Kendall A, Cipolla R (2017): Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 39(12), 2481-2495.
[14] Simonyan K, Zisserman A (2014): Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations*, San Diego, USA.
[15] Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M (2015): Imagenet large scale visual recognition challenge. *International Journal of Computer Vision,* 115(3), 211-252.
[16] Sharif Razavian A, Azizpour H, Sullivan J, Carlsson S (2014): CNN features off-the-shelf: an astounding baseline for recognition. *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Columbus, USA.
[17] Yosinski J, Clune J, Bengio Y, Lipson H (2014): How transferable are features in deep neural networks? *Neural Information Processing Systems 2014*, Montreal, Canada.
[18] Garcia-Garcia A, Orts-Escolano S, Oprea S, Villena-Martinez V, Garcia-Rodriguez J (2017): A review on deep learning techniques applied to semantic segmentation. *arXiv preprint arXiv:1704.06857*.
[19] Kuznetsova A, Rom H, Alldrin N, Uijlings J, Krasin I, Pont-Tuset J, Kamali S, Popov S, Malloci M, Duerig T (2018): The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv preprint arXiv:1811.00982*.
[20] Scikit-image (2019): *Histogram Equalization*. Retrieved from https://scikit-image.org/docs/dev/auto_examples/color_exposure/plot_equalize.html
[21] Otsu N (1979): A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics,* 9(1), 62-66.
[22] Ji X, Miao Z, Kromanis R (2020): Vision-based measurements of deformations and cracks for RC structure tests. *Engineering Structures*. [under review]
[23] Hartley R, Zisserman A. (2003): *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd edition.

