Smart Structures and Systems, *Vol. 31*, *No. 4* (*2023*) *325-334*
https://doi.org/10.12989/sss.2023.31.4.325

A hierarchical semantic segmentation framework for computer vision-based bridge damage detection

Jingxiao Liu*[1], Yujie Wei [2], Bingqing Chen [2] and Hae Young Noh [1]

[1] *Department of Civil and Environmental Engineering*, *Stanford University*, *Stanford*, *CA*, *USA*
[2] *Department of Civil and Environmental Engineering*, *Carnegie Mellon University*, *Pittsburgh*, *PA*, *USA*

(*Received September 13, 2022*, *Revised December 8, 2022*, *Accepted February 2, 2023*)

Abstract. Computer vision-based damage detection enables non-contact, efficient and low-cost bridge health monitoring, which reduces the need for labor-intensive manual inspection or that for a large number of on-site sensing instruments. By leveraging recent semantic segmentation approaches, we can detect regions of critical structural components and identify damages at pixel level on images. However, existing methods perform poorly when detecting small and thin damages (e.g., cracks); the problem is exacerbated by imbalanced samples. To this end, we incorporate domain knowledge to introduce a hierarchical semantic segmentation framework that imposes a hierarchical semantic relationship between component categories and damage types. For instance, certain types of concrete cracks are only present on bridge columns, and therefore the non-column region may be masked out when detecting such damages. In this way, the damage detection model focuses on extracting features from relevant structural components and avoid those from irrelevant regions. We also utilize multi-scale augmentation to preserve contextual information of each image, without losing the ability to handle small and/or thin damages. In addition, our framework employs an importance sampling, where images with rare components are sampled more often, to address sample imbalance. We evaluated our framework on a public synthetic dataset that consists of 2,000 railway bridges. Our framework achieves a 0.836 mean intersection over union (IoU) for structural component segmentation and a 0.483 mean IoU for damage segmentation. Our results have in total 5% and 18% improvements for the structural component segmentation and damage segmentation tasks, respectively, compared to the best-performing baseline model.

Keywords: bridge health monitoring; computer vision; damage detection; semantic segmentation

1. Introduction

Bridges are critical components of transportation infrastructure that connect people, roadways, railways, communities, etc. Yet around 46,154 bridges in the U.S. are structurally deficient (ASCE 2021); this calls for the development of efficient, cost-effective, and accurate bridge health monitoring approaches. Crack is one of the most common causes of bridge damage (Xu *et al*. 2019), and its detection is an important task for bridge health monitoring. Therefore, in this paper, we focus on detecting concrete cracks and exposed rebars on bridge columns.

Bridge conditions are currently monitored via manual inspection (Hartle *et al*. 2002) and sensor instrumentation (Spencer *et al*. 2019, Liu *et al*. 2020, 2022). For the majority of bridges, where sensors are not available, manual inspection by trained inspectors is the primary approach for structural health monitoring (SHM). However, manual inspection is labor-intensive, time-consuming, infrequent, and potentially dangerous (Sun *et al*. 2020). Sensor-based bridge health monitoring methods overcome some of the drawbacks of manual inspection methods, by continuously collecting structural performance data with sensing systems, but sensor-based methods are still costly and lack scalability as they require on-site installation and maintenance of equipment and instruments (Sony *et al*. 2019).

Recently, computer vision-based (CV-based) techniques, which use remote cameras and unmanned aerial vehicles (UAVs) to capture visual information of bridges, have gained popularity in bridge health monitoring. The low cost, efficiency, and accuracy of CV-based techniques hold the potential to streamline the current process of bridge health monitoring. A variety of computer vision techniques have been applied to recognize structural elements and to detect bridge damages (Spencer *et al*. 2019, Dong and Catbas 2021). In this paper, we focus on detecting concrete cracks and exposed rebars on bridge columns.

Existing CV-based damage detection approaches are broadly classified into two groups: object detection and semantic segmentation. Object detection-based approaches (Dung 2019, Kim and Cho 2019, Deng *et al*. 2020) treat damages, such as cracks, exposed rebars, and holes, as particular types of objects and aim to detect their presence on images. The output of an object detection approach could be a bounding box or an outline surrounding the detected damage. Such object detection-based approaches are capable of detecting small objects on a high-resolution image, making them robust to view changes. However, object detection-based approaches could only output a rough outline of the damage, which is insufficient to

*Corresponding author, Ph.D. Candidate,
E-mail: liujx@stanford.edu

ISSN: 1738-1584 (Print), 1738-1991 (Online)

support further analysis on the damage, such as evaluating the width, length, and growing speed of a crack. In contrast, semantic segmentation approaches (Liu *et al*. 2019, Ren *et al*. 2020, Zhang *et al*. 2019, Bang *et al*. 2019) take an image as an input and directly label each pixel based on its state (i.e., multiple damage states). Semantic segmentation-based approaches could provide a fine-granular annotation of damages on an image. Thus, semantic segmentation is adopted in this study.

Applying existing semantic image segmentation methods to detect bridge damage is still challenging due to the following two reasons:

(1) Small and thin object challenge: Cracks and exposed rebars are small and thin objects. Unlike traditional semantic components (e.g., columns, beams, cars, and people), these structural damage objects do not have regular shapes and usually occupy a small fraction of the entire object. Thus, the pixel-level damage segmentation is more challenging than traditional semantic segmentation.
(2) Imbalanced data challenge: There are two types of data imbalance worth noticing: compared to regular components, the total number of images containing damaged components only account for a small portion of all captured images (sample imbalance), and the segment of damaged components only account for a small area of a complete image (area imbalance). Both types of data imbalance create challenges when detecting and segmenting damaged components on images.

To overcome the above challenges, this paper introduces a hierarchical semantic segmentation framework based on domain knowledge for bridge damage segmentation. The framework contains two modules: 1) a component semantic segmentation module that predicts the category of bridge components (beams, columns, rails, sleepers, etc.) for each pixel using raw images as input and 2) a damage detection module that predicts the category of damage types (concrete cracks, exposed rebars, etc.) for each pixel using both raw images and the predicted component semantics as input. Instead of treating semantic information of structural components and damage separately, the proposed framework leverages the domain knowledge that the structural damage of interest may only reside on specific structural components (e.g., diagonal and splitting cracks mainly occur on columns due to their insufficient load carrying capacity or inadequate cross-section) and propagates the semantic information on the structural components when detecting such damages. Specifically, we first predict structural components and mask out non-column regions in the original images. In this way, the proposed framework could partially address the small and thin object challenge by focusing only on the possible damage regions using component semantics. During training and testing, we also conduct a multi-scale augmentation that resizes the original image with different ratios. This framework provides the segmentation model with various zoom-in and zoom-out images, which shows closer views of small cracks and preserves contextual information of the entire structure. Furthermore, to address the challenge from imbalanced data, we introduce an importance sampling method for both the structural component and damage segmentation tasks. This method repeatedly samples images containing rare components (e.g., railway sleepers and exposed rebars) to make the dataset less imbalanced.

We evaluated the proposed framework using a publicly available synthetic dataset - the Tokaido dataset (Narazaki *et al*. 2021) - that includes images taken from 2,000 railway bridges. We tested our framework for three tasks: 1) bridge component segmentation, 2) damage segmentation on extracted pure-texture images 3) damage segmentation on bridge images. In summary, our framework achieves a 0.836 mean intersection over union (IoU) for the structural component segmentation task, a 0.712 mean IoU (mIoU) for the pure texture damage segmentation task, and a 0.483 mean IoU for the real scene damage segmentation task. The experiment results highlighted that the proposed framework outperformed the baseline ensembled models which do not have the semantic-guided damage detection, importance sampling, and multi-scale augmentation.

2. Background of CV-based bridge damage detection and semantic image segmentation

This section provides a review of existing CV-based bridge damage detection methods and semantic image segmentation methods to identify research challenges and highlight the contributions of this study.

2.1 Computer-vision-based bridge damage detection

CV-based techniques can automate part of the current laborious and costly process for manual vision inspection, for tasks, such as structural component recognition and damage detection, The early work in CV-based bridge damage detection focused on extracting hand-crafted features defined based on engineering judgment. For instance, edge detection filters were applied to identify concrete cracking (Abdel-Qader *et al*. 2003). However, these approaches depend on damage type-specific engineering and fine-tuning, which constrain their scalability and applicability to potentially complex real-world scenarios (Spencer *et al*. 2019).

Given the tremendous success of convolutional neural networks (CNN), it has become increasingly popular in recent years. Both structural component recognition (Liang 2019) and damage detection (Cha *et al*. 2017) may be formulated as an object detection task, i.e., finding bounding boxes around the structural components/damage. In comparison, semantic segmentation, which classifies each pixel in an image, provides fine-granular information, e.g., the precise location of the damage. Specifically, we are inspired by Hoskere *et al*. (2018), which considers structural information jointly with damage detection; they used three different neural networks to simultaneously classify semantic information, presence of damage, and damage types as a multi-task learning problem. In this

work, we adopt a hierarchical architecture and uses semantic information on structural components as input to the downstream damage segmentation task.

2.2 Semantic image segmentation

Both the scene component recognition and damage detection tasks could be formed as a semantic segmentation problem. Semantic segmentation, an important task towards scene understanding, is the process of labeling each pixel on an image based on its semantics. The following paragraphs provide an overview of existing semantic segmentation methods and provide technical background to the proposed framework.

In general, semantic segmentation can be decomposed into two subtasks: feature extraction (encoder) and segmentation reconstruction (decoder) (Ronneberger *et al*. 2015, Shelhamer *et al*. 2017). The encoder extracts feature from an image to form a low-dimensional representation of integrated knowledge of the image content. Example encoders include convolutional neural networks (CNN), dilated convolutions, feature fusion, multi-scale prediction, recurrent neural networks, and transformers. The decoder takes an abstract feature vector as input and reconstructs a low-dimensional feature into a high-dimensional semantic segmentation image. Typical decoders include fully connected networks, feature pyramid networks, and object contextual representations (Noh *et al*. 2015, Zhao *et al*. 2017, Yuan *et al*. 2020).

In the proposed framework, we use a combination of three different types of encoders. Below is a detailed introduction to each of them.

(1) HRNet (Wang *et al*. 2020): Hi-Resolution Net is a widely convolutional neural network for extracting features for object recognition and semantic segmentation on high-resolution images. On the one hand, the model leverages conventional convolution layers to gradually convert high-resolution images into low-resolution features. On the other hand, the HRNet model leverages the multi-resolution connection to preserve detailed information that existed only in high-res textures.
(2) ResNest (Zhang *et al*. 2022): ResNest is a representative feature extractor from the ResNet family that leverages residuals to address the vanishing gradients when training a deep neural network. Compared to the conventional ResNet, ResNest introduces the split-attention block to take advantage of channel-wise attention when extracting features while preserving the cross-channel feature correlations.
(3) Swin (Liu *et al*. 2021): Swin Transformer is an attention-based encoder that serves as a general-purpose backbone for semantic segmentation. As transformer-based methods gained huge success in the natural language processing domain, visual-transformers outperformed conventional CNN-based backbones on many public datasets. By introducing sliding windows in a hierarchical way, Swin Transformer could leverage the attention strategy when computing local features without introducing too much extra computation overhead.

The proposed framework also uses two decoders as described below:

(1) PSPNet (Zhao *et al*. 2017): The pyramid scene parsing decoder is known for its capability of extracting global context information for reconstructing semantic segmentation by using pyramid pooling. Similar to image feature pyramids, PSP uses a pyramid of convolution layers that performs pooling on image crops at different scales. On the one hand, a large pooling layer allows the network to extract context information with a large perception field. On the other hand, a small pooling layer enables the use of local features for semantic understanding. In the proposed framework, we used PSPNet together with ResNest and Resnet to create our models.
(2) OCR (Yuan *et al*. 2020): Object-contextual representation is a decoder that aims at improving semantic segmentation accuracy by introducing soft object regions as part of the input of the network. Based on a rough semantic segmentation, the decoder first computes an object region representation, then classifies each pixel based on their relationships with the object region representations. The use of the intermediate object region representations enhances the concept of an "object" instead of labeling each pixel independently.
(3) UperNet (Xiao *et al*. 2018): UperNet is a unified perceptual parsing network that could be used to parse various visual concepts at the same time, such as scene, objects, parts, and textures, which makes it suitable for the task in this paper which requires semantic segmentation and damage segmentation at the same time. UperNet is built on top of the feature pyramid network and pyramid pooling module but uses precise ROI pooling rather than adaptive pooling for its pyramid pooling module. The details of UperNet implementation can be found in the original paper.

In our framework, the final ensemble model comes from different combinations of the encoders and decoders above, including 1) HRNet + PSPNet; 2) ResNest + PSPNet; 3) HRNet + OCRNet; 4) ResNest+UperNet; and 5) Swin + UPerNet. The following section provides a more detailed introduction to our segmentation framework.

3. Hierarchical semantic segmentation framework

This section introduces the details of the proposed hierarchical semantic segmentation framework (as shown in Fig. 1) that integrates the structural component segmentation and damage segmentation together to achieve accurate damage detection. The framework contains two modules: 1) a component semantic segmentation module that predicts the category of bridge components (beams,

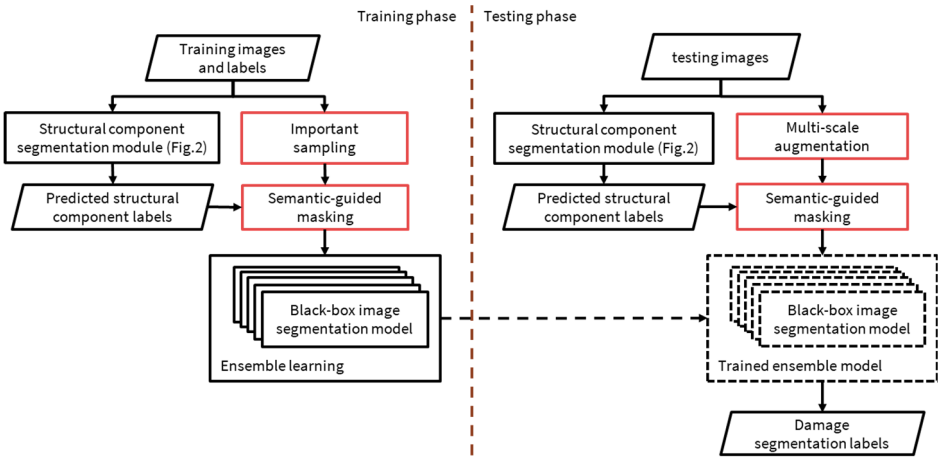


Fig. 1 Flowchart of our hierarchical semantic segmentation framework for bridge column crack detection

columns, rails, sleepers, etc.) for each pixel using raw images as input and 2) a damage detection module that predicts the category of damage types (concrete cracks, exposed rebars, etc.) for each pixel using both raw images and the predicted component semantics as input. Specifically, our implementation utilizes the semantic hierarchy between columns and column damages. It eliminates the influence of noncritical regions by masking out non-column pixels in the original images, forcing the damage segmentation model to focus only on the regions that have damages.

We first provide an overview of the proposed framework. Previous studies have shown that hierarchical segmentation (Li *et al*. 2022, Zhou *et al*. 2019) and importance sampling (Katharopoulos and Fleuret 2018, Shrivastava *et al*. 2016) could be used for addressing small object and imbalanced data problems. Therefore, the workflow of our framework focuses on detecting small damages by leveraging two heuristics: 1) structural damages only present on structural components; 2) structural damages can have various sizes and directions shown on images. The first heuristic leads to a hierarchical detection workflow which aims at segmenting structural components first and then detecting structural damages on structural component segments. The second heuristic leads to an importance sampling and multi-scale augmentation method that repeatedly sample images containing structural damages with various views and sizes.

Specifically, in the training phase, images are input into the importance sampling module and the structural component segmentation module that is learned following the flowchart in Fig. 2. The importance sampling module more frequently samples images containing rare classes (e.g., exposed rebar) to address the imbalanced data challenge. The structural component segmentation module predicts component segments. Then, the importantly sampled data and column segment labels of each image are input to the next module that masks out non-column regions, which is to impose a hierarchical semantic relationship between columns and column damages. The masked images and corresponding labels are used to train the black-box image segmentation models in the ensemble learning module.

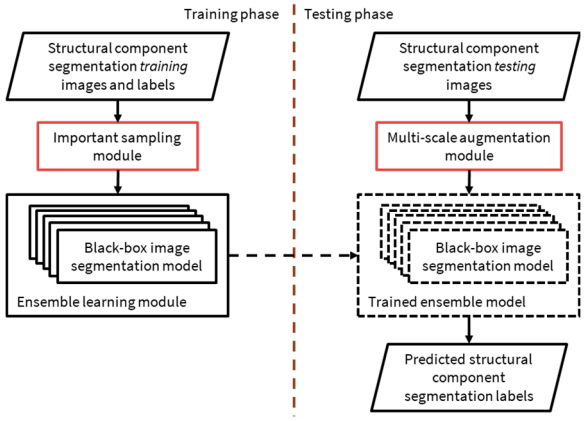


Fig. 2 Flowchart of our structural component segmentation module

During the testing phase, testing images are first input into the multi-scale augmentation module that augments the original images with different zoom ratios, which provides closer views of damages to overcome the small and thin objects challenge and retains contextual information of the entire structure. After masking out non-column components, we use the trained ensemble model to predict damage segmentation labels. The following subsections provides more details of the proposed framework from five perspectives: 1) bridge component semantic segmentation; 2) importance sampling; 3) semantic-guided masking; 4) ensemble learning and 5) multi-scale augmentation.

3.1 Bridge component semantic segmentation

The workflow of bridge component semantic segmentation is shown in Fig. 2. During the training phase, bridge component images together with their semantic segmentation (Non-bridge, Slab, Beam, Column, Nonstructural, Rail, Sleeper, Other) are used for training a set of deep learning models. During the testing phase, the trained model could take a bridge scene image as input and predict its semantic segmentation. Notice that the workflow

also includes importance sampling, multi-scale augmentation, and ensemble learning. The mechanisms of these three features, which are described in the following subsections, are in general the same as those in the damage detection framework shown in Fig. 1. Specifically, for the bridge component semantic segmentation task, the importance sampling is conducted on railway components (e.g., rails and sleepers) as they are more difficult to recognize due to their thin shape and rare occurrence in the dataset.

3.2 Importance sampling

As mentioned above, importance sampling is employed to repeatedly samples images containing components of interest to overcome the imbalanced data challenge. The importance weights are calculated based on class occurrence, i.e., the less a particular class of component presents in the dataset, the higher importance weight it has during resampling. For example, in the implementation the sleeper class was assigned a much higher importance weight because only around 6% of images have sleeper components in the training dataset. The importance sampling algorithm contains two hyper-parameters: 1) n_m, the minimum number of pixels of the rare class in the image, and 2) r, the number of repeating. We first count the number of pixels of the important component in each image defined as n. Then, we repeated sample r times if $n > n_m$. The two hyper-parameters for importance sampling are selected through fine-tuning in the evaluation step.

3.3 Semantic-guided masking

Semantic-guided masking bridges the bridge component semantic segmentation module and the damage detection module. Given an image as input, the bridge component recognition module first computes the semantic segmentation of bridge components. The raw image together with the semantic segmentation are passed to the damage detection module. Since column crack damages could only reside on bridge columns, the damage detection module leverages semantic information to mask out pixels that do not belong to column segments in each image, which allows the model to focus on distinguishing damage vs. non-damage regions. Specifically, the semantic-guided masking replaces the pixels that are non-column with black RGB values (i.e., [0, 0, 0]). Fig. 3. shows an example of the original image and the masked image.

Fig. 3 An example of our semantic-guided masking module. Non-column regions are replaced with black

3.4 Ensemble learning

To reduce the variance due to model training and data sampling, we use a simple majority vote as a way to ensemble machine learning models for both baseline approaches and the proposed framework in the implement-tations. Specifically, we conduct majority voting over predictions using multiple image segmentation methods in order to improve the final prediction accuracy. For both the baseline approach and the proposed method, we first trained five image semantic segmen tation models indivisually (HRNet + PSPNet, ResNest + PSPNet, HRNet + OCRNet, ResNest+UperNet, and Swin + UPerNet) as introduced in the background section. Then, the final prediction of each pixel is the mode of the predictions of the five segmentation models.

3.5 Test-time multi-scale augmentation

Despite the fact that the data has been augmented during the training phase (flip, crop, scale, photo-metric distortion, etc.), the semantic segmentation and damage detection modules are still susceptible to scale changes during the testing phase. Therefore, multi-scale augmentation is also employed in the testing phase to make multiple predictions of the same region of an image using crops with different scales as input. Our segmentation models are trained and tested with the same size of image patches cropped from each input image. By resizing the original image with different ratios and sliding the crop window across the image, the segmentation model could make predictions using both detailed textures of thin objects (e.g., railway sleepers and damages) and preserve context information of components and surrounding objects. Fig. 4 shows an example of the test-time multi-scale augmentation.

4. Evaluation

In this section, we evaluated the proposed approach for structural component semantic segmentation and damage segmentation using the Tokaido dataset (Narazaki *et al.* 2021). Below first describes the dataset used for evaluation and the implementation details of the proposed approach, followed by evaluation results on the Tokaido dataset and the discussions on the results.

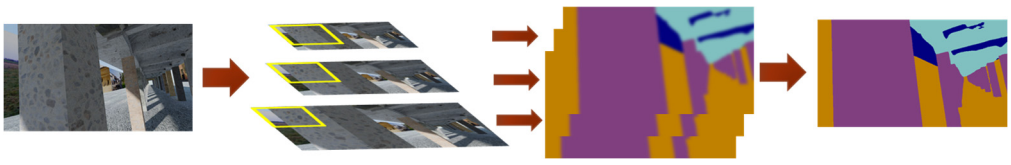
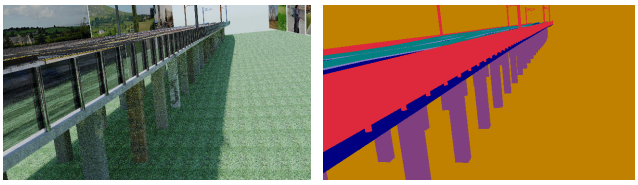
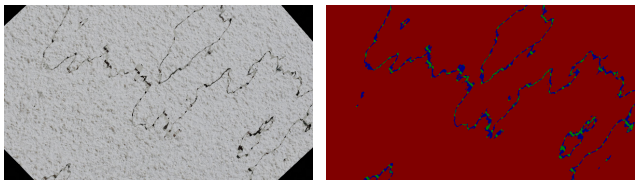


Fig. 4 An example of the test time multiscale augmentation module. The final prediction ensembles predictions of the same image with multiple zoom ratios



(a) Synthetic image (b) Ground truth

Fig. 5 An example of (a) synthetic image for structural components segmentation; and (b) ground truth structural components labels



(a) Synthetic image (b) Ground truth

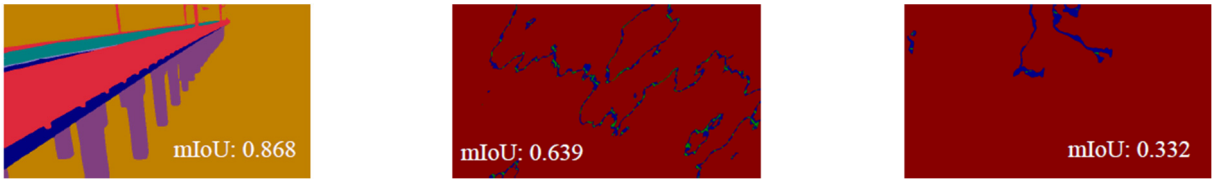
Fig. 6 An example of (a) synthetic image for pure texture damage segmentation; and (b) ground truth damage labels

(a) Synthetic image (b) Ground truth

Fig. 7 An example of (a) synthetic image for real scene damage segmentation; and (b) ground truth damage labels

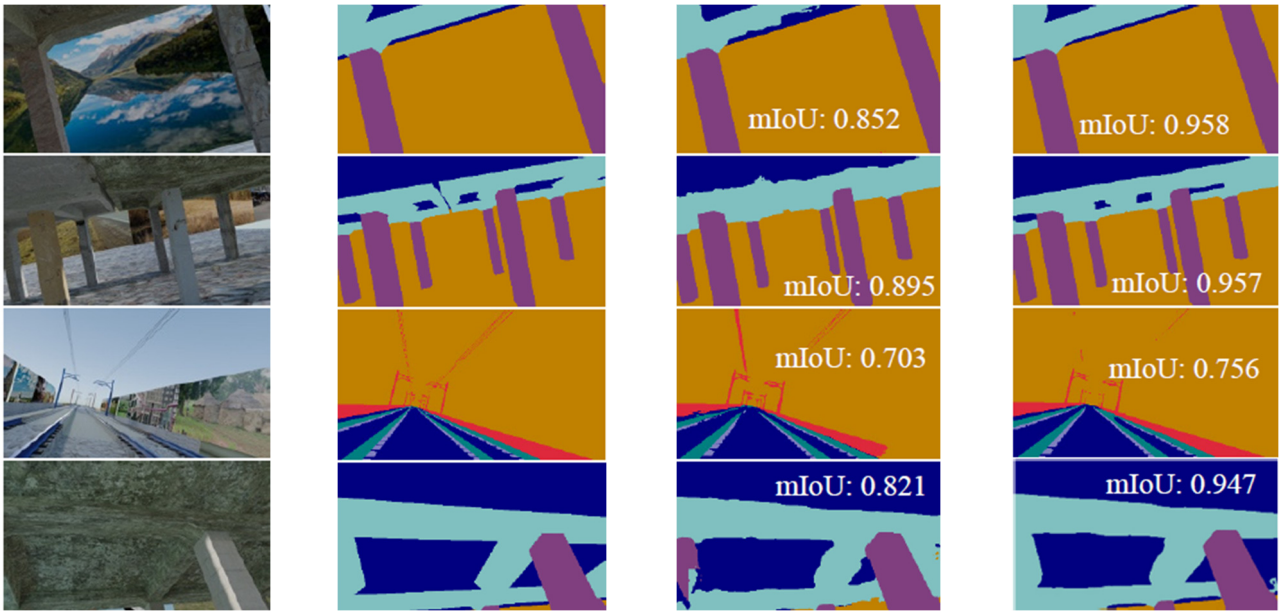
4.1 Dataset description

The Tokaido dataset is a publicly available synthetic image dataset that consists of 2,000 railway bridges with random geometry realized by the actual design procedure. In the dataset, random damages, including concrete cracks and exposed rebar, could be present on the bridge columns. In our evaluation, images of 1,750 bridges are used for training and validation, and images of the other 250 bridges are used for testing. The dataset supports evaluations from



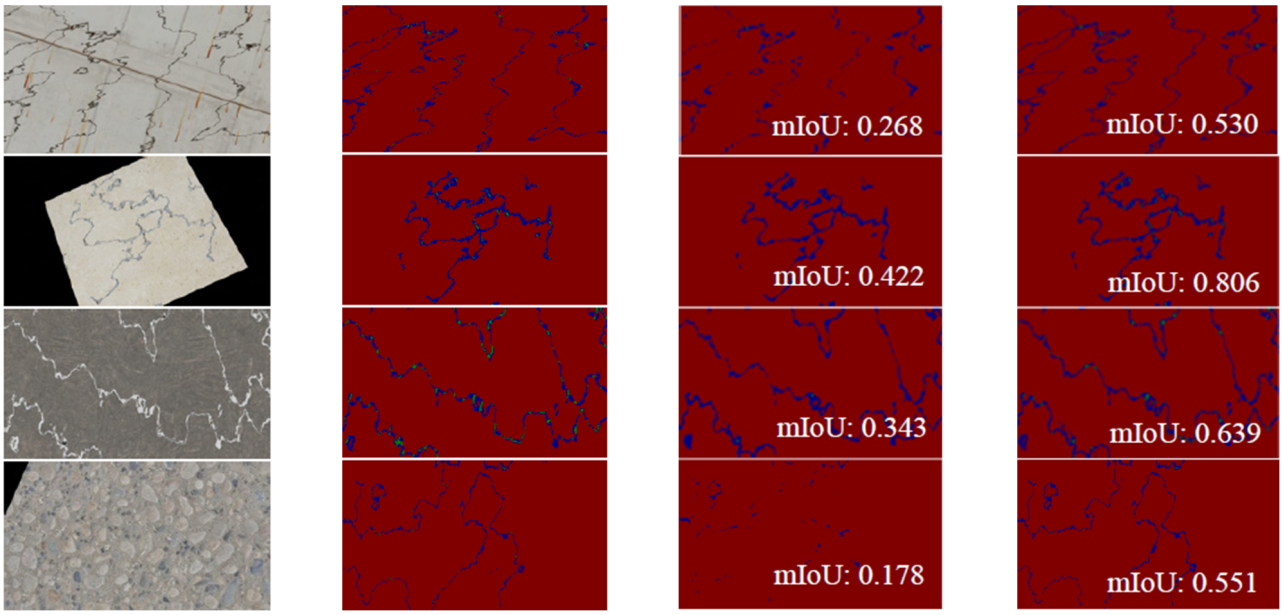
(a) Structural component (b) Damage segmentation (c) Pure texture damage segmentation

Fig. 8 (a) Structural component and (b), (c) damage segmentation results for the examples in Figs. 5-7



(a) Image (b) Ground truth (b) Baseline (c) Ours

Fig. 9 Example semantic segmentation results between the baseline and the proposed framework



(a) Image (b) Ground truth (b) Baseline (c) Ours

Fig. 10 Example damage segmentation results between the baseline and the proposed framework

three perspectives: 1) structural components segmentation, 2) damage segmentation for real scene images, and 3) damage segmentation for pure texture images. For the structural component segmentation task, there are 8,648 images (as shown in Fig. 5) with seven classed of components: non-bridge, slab, beam, column, non-structural, rail, and sleeper. For the damage segmentation tasks, the dataset has 7,990 real scene images (as shown in Fig. 6) and 2,700 pure texture images (as shown in Fig. 7) with three classes: non-damage, concrete damage, and exposed rebar.

4.2 Implementation details

This subsection introduces the experimental setup of our hierarchical semantic segmentation framework. Specifically, to overcome the imbalanced data challenge, we first conduct importance sampling by repeatedly sampling images containing railway components and exposed rebar damages ten times. Then, we split the sampled dataset into a 90% training set and a 10% validation set. One should note that the splits for structural components and real scene damage segmentation tasks are based on bridges, i.e., we split bridges geometry into two random subsets in a ratio of 9:1 and use images taken from training bridges subset for model training and others for model validation. Furthermore, each semantic segmentation model is trained with two 16 GB Nvidia Tesla P100 GPUs and optimized using a stochastic gradient descent optimizer with a polynomial learning rate schedule, in which the learning rate is decayed to a polynomial function. Hyper-parameters, such as learning rate, batch size, and crop size, are selected via fine-tuning and cross-validation. Specifically, the polynomial decay schedule has an initial learning rate of 0.01, an end learning rate of 0.0001, and a power of the polynomial of 0.9. The batch size is 16. The batch size is 16. The crop size is 640 × 360. Other implementation details can be found in the code repository: https://github.com/jingxiaoliu/bridge-damage-segmentation.

4.3 Results and discussion

This subsection presents our evaluation results and discussions of the findings. Intersection over Union (IoU) is used as the primary metric to evaluate the model performance. IoU is defined as the area of intersection between the predicted segmentation and the ground truth segmentation over that of the union. When the predicted segmentation aligns with the ground truth perfectly, IoU is 1. When the predicted segmentation has no overlapping region with the ground truth segmentation, the IoU is 0.

Fig. 8 shows examples of the structural component and damage segmentation results predicted using our framework. Compared to the corresponding ground truth labels in Figs. 5-7, we can observe that our framework successfully predicts structural components and damaged regions. Figs. 9-10 further compares the semantic segmentation and damage detection performance between the proposed framework and the baseline. As shown in the Fig. 9, after employing the proposed framework, it can be seen that the segmentation of sleepers is continuous and clear while maintaining the accuracy of non-structural and rail components. The segmentation of structural components, such as slabs, beams, and columns also become more accurate. Moreover, Fig. 10 provides a set of example results for the damage detection task. The results highlight that the proposed framework could better detect small damages such as exposed rebars and discontinuous cracks compared to the baseline. Besides the intuitive visualization, below gives a quantitative comparison between the baseline that leverages ensembled learning on the state-of-the-art models and the proposed framework that further exploits

Table 1 IoUs for structural component segmentation. ENS stands for the ensembled model. I.S. and M.S. mean training with importance sampling and testing with multi-scale, respectively

	Slab	Beam	Column	Non-structural	Rail	Sleeper	Average
ENS (baseline)	0.891	0.880	0.859	0.660	0.623	0.701	0.785
ENS+I.S.	0.915	0.912	0.958	0.669	0.618	0.892	0.827
ENS+I.S.+M.S. (ours)	0.924	0.929	0.965	0.681	0.621	0.894	0.836

Table 2 IoUs for damage segmentation of pure texture images

	Concrete damage	Exposed rebar	Average
ENS (baseline)	0.356	0.536	0.446
ENS+I.S.	0.708	0.714	0.711
ENS+I.S.+M.S. (ours)	0.698	0.727	0.712

Table 3 IoUs for damage segmentation of real scene images. S.G.M. means semantic-guided masking

	Concrete damage	Exposed rebar	Average
ENS (baseline)	0.235	0.365	0.300
ENS+I.S.	0.340	0.557	0.448
ENS+I.S.+M.S.	0.350	0.583	0.467
ENS+I.S.+M.S.+S.G.M. (ours)	0.379	0.587	0.483

the power of semantic-guided mask and importance sampling.

Tables 1-3 summarize the IoU results for structural component segmentation, damage segmentation of pure texture images, and damage segmentation of real scene images, respectively. For all the three segmentation tasks, the proposed approach (ENS + I.S. + M.S. and ENS + I.S. + M.S. + S.G.M.) outperformed baseline approaches that do not use the importance sampling and/or multi-scale testing. Specifically, for the structural component segmentation task, the importance sampling (I.S.) and multi-scale testing (M.S.) improved the mean IoU by 4% and 1%, respectively; for the damage segmentation of pure texture tasks, I.S. and M.S. improved the mean IoU by 25% and 0.1%, respectively; and for the damage segmentation of real scene images task, I.S. and M.S. improved the mean IoU by 15% and 2%, respectively. In addition, for damage segmentation of real scene images, our framework that conducts semantic-guided masking (S.G.M.) improved the damage detection results by 1.6%. Furthermore, when testing on the 250 testing bridge data, our framework achieves a 0.832 mean IoU for structural component segmentation, and a 0.441 mean IoU for damage segmentation.

The result tables show that the importance sampling has the largest improvement (around a 20% increase of IoU) on segmenting railway sleepers and exposed rebar damages. These improvements show the effectiveness of importance sampling on recognizing classes that have less occurrence and smaller area. Furthermore, the IoU results for predicting non-structural, rail components, and damages increase after conducting multi-scale testing, which validates the effectiveness of our framework for accurately predicting small and thin objects.

5. Conclusions

In this work, we used a synthetic dataset to train a semantic segmentation model for scene understanding of civil infrastructures and detecting structural damages. The proposed framework addressed three key challenges when dealing with infrastructure scenes. Firstly, the proposed framework improved scene understanding and damage detection accuracy by resampling components of interest that are rarely presented in images. The experiment results highlighted that the resampling technique addressed the category imbalance problem existing both across different data points and within a particular data point. Secondly, the proposed framework handles the perspective view changes using test-time multi-scale augmentation. Considering the fact that objects of interest may have different scales in the captured images, the proposed framework recognizes scene components and damages using crops with different scales at testing time to mitigate the variation in viewpoint. Finally, the proposed damage detection module leverages domain knowledge, such as concrete damages could only occur on concrete components to further refine the damage detection results. By masking the non-concrete regions when feeding an image to the damage detection network, the proposed framework could use the predicted semantics to guide the damage detection, thereby improving the prediction accuracy and reducing the computation time. The experiment results discussed in the previous section have validated the feasibility and the effectiveness of the proposed framework.

Note that in this study it is assumed that semantic segmentation labels are available for the training set. Given the practical difficulty of collecting and annotating a sufficiently large dataset for deep learning from real-world scenes, we envision using either simulation-to-real transfer learning from synthetically generated data or self-supervised learning on unlabeled data as pathways for our current work to be transferred to practical applications.

Acknowledgments

This research was supported in part by the Leavell Fellowship on Sustainable Built Environment at Stanford University.

References

Abdel-Qader, I., Abudayyeh, O. and Kelly, M.E. (2003), "Analysis of edge-detection techniques for crack identification in bridges", *J. Comput. Civil Eng.*, **17**(4), 255-263. https://doi.org/10.1061/(ASCE)0887-3801(2003)17:4(255)

ASCE (2021), Bridges-infrastructure report card.

Bang, S., Park, S., Kim, H. and Kim, H. (2019), "Encoder–decoder network for pixel-level road crack detection in black-box images", *Comput.-Aided Civil Infrastr. Eng.*, **34**(8), 713-727. https://doi.org/10.1111/mice.12440

Cha, Y.J., Choi, W. and Büyüköztürk, O. (2017), "Deep learning-based crack damage detection using convolutional neural networks", *Comput.-Aided Civil Infrastr. Eng.*, **32**(5), 361-378. https://doi.org/10.1111/mice.12263

Deng, J., Lu, Y. and Lee, V.C.S. (2020), "Concrete crack detection with handwriting script interferences using faster region-based convolutional neural network", *Comput.-Aided Civil Infrastr. Eng.*, **35**(4), 373-388. https://doi.org/10.1111/mice.12497

Dong, C.Z. and Catbas, F.N. (2021), "A review of computer vision–based structural health monitoring at local and global levels", *Struct. Health Monitor.*, **20**(2), 692-743. https://doi.org/10.1177/1475921720935585

Dung, C.V. (2019), "Autonomous concrete crack detection using deep fully convolutional neural network", *Automat. Constr.*, **99**, 52-58. https://doi.org/10.1016/j.autcon.2018.11.028

Hoskere, V., Narazaki, Y., Hoang, T.A. and Spencer Jr, B.F. (2018), "Towards automated post-earthquake inspections with deep learning-based condition-aware models", *arXiv preprint* arXiv:1809.09195.

Hartle, R.A., Ryan, T.W., Mann, E., Danovich, L.J., Sosko, W.B., Bouscher, J.W. and Baker Jr, M. (2002), "Bridge Inspector's Reference Manual: Volume 1 and Volume 2 (No. DTFH61-97-D-00025)", United States. Federal Highway Administration.

Katharopoulos, A. and Fleuret, F. (2018), "Not all samples are created equal: Deep learning with importance sampling", *Proceedings of International Conference on Machine Learning*, pp. 2525-2534.

Kim, B. and Cho, S. (2019), "Image-based concrete crack assessment using mask and region-based convolutional neural network", *Struct. Control Health Monitor.*, **26**(8), e2381. https://doi.org/10.1002/stc.2381

Li, L., Zhou, T., Wang, W., Li, J. and Yang, Y. (2022), "Deep Hierarchical Semantic Segmentation", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1246-1257.

Liang, X. (2019), "Image-based post-disaster inspection of reinforced concrete bridge systems using deep learning with Bayesian optimization", *Comput.-Aided Civil Infrastr. Eng.*, **34**(5), 415-430. https://doi.org/10.1111/mice.12425

Liu, Z., Cao, Y., Wang, Y. and Wang, W. (2019), "Computer vision-based concrete crack detection using U-net fully convolutional networks", *Automat. Constr.*, **104**, 129-139. https://doi.org/10.1016/j.autcon.2019.04.005

Liu, J., Chen, S., Bergés, M., Bielak, J., Garrett, J.H., Kovačević, J. and Noh, H.Y. (2020), "Diagnosis algorithms for indirect structural health monitoring of a bridge model via dimensionality reduction", *Mech. Syst. Signal Process.*, **136**, 106454. https://doi.org/10.1016/j.ymssp.2019.106454

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S. and Guo, B. (2021), "Swin transformer: Hierarchical vision transformer using shifted windows", *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10012-10022.

Liu, J., Xu, S., Bergés, M. and Noh, H.Y. (2022), "HierMUD: Hierarchical multi-task unsupervised domain adaptation between bridges for drive-by damage diagnosis", *Struct. Health Monitor*, p. 14759217221081159. https://doi.org/10.1177/14759217221081159

Narazaki, Y., Hoskere, V., Yoshida, K., Spencer Jr., B.F. and Fujino, Y. (2021), "Synthetic environments for vision-based structural condition assessment of Japanese high-speed railway viaducts", *Mech. Syst. Signal Process.*, **160**, 107850. https://doi.org/10.1016/j.ymssp.2021.107850

Noh, H., Hong, S. and Han, B. (2015), "Learning deconvolution network for semantic segmentation", *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1520-1528.

Ren, Y., Huang, J., Hong, Z., Lu, W., Yin, J., Zou, L. and Shen, X. (2020), "Image-based concrete crack detection in tunnels using deep fully convolutional networks", *Constr. Build. Mater.*, **234**, 117367. https://doi.org/10.1016/j.conbuildmat.2019.117367

Ronneberger, O., Fischer, P. and Brox, T. (2015), "U-net: Convolutional networks for biomedical image segmentation", *Proceedings of International Conference on Medical image computing and computer-assisted intervention*, pp. 234-241.

Shelhamer, E., Long, J. and Darrell, T. (2017). "Fully convolutional networks for semantic segmentation", *IEEE Trans. Pattern Anal. Mach. Intell.*, **39**(4), 640-651.

Shrivastava, A., Gupta, A. and Girshick, R. (2016), "Training region-based object detectors with online hard example mining", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 761-769.

Sony, S., Laventure, S. and Sadhu, A. (2019), "A literature review of next-generation smart sensing technology in structural health monitoring", *Struct. Control Health Monitor.*, **26**(3), e2321. https://doi.org/10.1002/stc.2321

Spencer Jr, B.F., Hoskere, V. and Narazaki, Y. (2019), "Advances in computer vision-based civil infrastructure inspection and monitoring", *Engineering*, **5**(2), 199-222. https://doi.org/10.1016/j.eng.2018.11.030

Sun, L., Shang, Z., Xia, Y., Bhowmick, S. and Nagarajaiah, S. (2020), "Review of bridge structural health monitoring aided by big data and artificial intelligence: From condition assessment to damage detection", *J. Struct. Eng.*, **146**(5), 04020073. https://doi.org/10.1061/(ASCE)ST.1943-541X.0002535

Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X. and Liu, W. (2020), "Deep high-resolution representation learning for visual recognition", *IEEE Transact. Pattern Anal. Mach. Intell.*, **43**(10), 3349-3364. https://doi.org/10.1109/TPAMI.2020.2983686

Xiao, T., Liu, Y., Zhou, B., Jiang, Y. and Sun, J. (2018), "Unified perceptual parsing for scene understanding", *Proceedings of the European conference on computer vision* (*ECCV*), pp. 418-434.

Xu, H., Su, X., Wang, Y., Cai, H., Cui, K. and Chen, X. (2019), "Automatic bridge crack detection using a convolutional neural network", *Appl. Sci.*, **9**(14), 2867. https://doi.org/10.3390/app9142867

Yuan, Y., Chen, X. and Wang, J. (2020), "Object-contextual representations for semantic segmentation", *Proceedings of European Conference on Computer Vision*, pp. 173-190.

Zhang, J., Lu, C., Wang, J., Wang, L. and Yue, X.G. (2019), "Concrete cracks detection based on FCN with dilated convolution", *Appl. Sci.*, **9**(13), 2686. https://doi.org/10.3390/app9132686

Zhang, H., Wu, C., Zhang, Z., Zhu, Y., Lin, H., Zhang, Z., Sun, Y., He, T., Mueller, J., Manmatha, R. and Li, M. (2022), "Resnest: Split-attention networks", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2736-2746.

Zhao, H., Shi, J., Qi, X., Wang, X. and Jia, J. (2017), "Pyramid scene parsing network", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2881-2890.

Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A. and Torralba, A. (2019), "Semantic understanding of scenes through

the ade20k dataset", *Int. J. Comput. Vision*, **127**(3), 302-321.
https://doi.org/10.1007/s11263-018-1140-0