Check for updates

Received: 7 December 2023 | Accepted: 7 April 2024

DOI: 10.1111/mice.13211

COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING WILEY

RESEARCH ARTICLE

Transformer-based framework for accurate segmentation of high-resolution images in structural health monitoring

M. Azimi | **T. Y. Yang**

Department of Civil Engineering, The University of British Columbia, Vancouver, British Columbia, Canada

Correspondence
T. Y. Yang, Department of Civil Engineering, The University of British Columbia, Vancouver, BC V6T 1Z4, Canada.
Email: yang@civil.ubc.ca

Abstract

High-resolution image segmentation is essential in structural health monitoring (SHM), enabling accurate detection and quantification of structural components and damages. However, conventional convolutional neural network-based segmentation methods face limitations in real-world deployment, particularly when handling high-resolution images producing low-resolution outputs. This study introduces a novel framework named Refined-Segment Anything Model (R-SAM) to overcome such challenges. R-SAM leverages the state-of-the-art zero-shot SAM to generate unlabeled segmentation masks, subsequently employing the DEtection Transformer model to label the instances. The key feature and contribution of the R-SAM is its refinement module, which improves the accuracy of masks generated by SAM without the need for extensive data annotations and fine-tuning. The effectiveness of the proposed framework was assessed through qualitative and quantitative analyses across diverse case studies, including multiclass segmentation, simultaneous segmentation and tracking, and 3D reconstruction. The results demonstrate that R-SAM outperforms state-of-the-art convolution neural network-based segmentation models with a mean intersection-over-union of 97% and a mean boundary accuracy of 87%. In addition, achieving high coefficients of determination in target-free tracking case studies highlights its versatility in addressing various challenges in SHM.

1 | INTRODUCTION

Traditional structural health monitoring (SHM) models rely on physics-based models with limited capabilities and may not be suitable for processing large volumes of data (e.g., using signal processing; Qarib & Adeli et al., 2016). On the other hand, data-driven models using machine learning provide versatile solutions; thus, they have been at the center of attention during the past few years. Deep learning (DL) has been used to develop end-to-end classification, detection, and segmentation tasks in SHM (Azimi & Pekcan et al., 2020). Most of the recent DL-based algorithms for SHM utilize convolution neural networks (CNNs), which have unparalleled performance in extracting local information but are limited in capturing long-range relationships between the features (Dosovitskiy et al., 2020; Sajedi & Liang et al., 2020). Therefore, CNN-based models perform poorly in extracting global contextual features from high-resolution images (e.g., damage detection under occlusion).

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.
© 2024 The Authors. *Computer-Aided Civil and Infrastructure Engineering* published by Wiley Periodicals LLC on behalf of Editor.

On the other hand, the resolution of portable cameras has increased significantly in the last few years. Due to the computational cost of processing high-resolution image data for DL, most recent computer vision models in SHM are designed and trained for low-resolution input data. Typical CNN-based models may not be suitable for high-resolution data because larger receptive fields are needed to capture the entire segmented regions. One approach that has been used in recent years is training a DL model with downsampled images, where the details are removed. Other methods have introduced new models using cropped images, which damages the context of the images (Azimi et al., 2020). Furthermore, fitting 4K image segmentation models into GPU memory is still challenging for typical computers. Even with stat-of-the-art high-performance computers, properly annotated high-resolution images are scarce for SHM tasks.

The earlier generation of CNN-based semantic segmentation models may achieve a high accuracy level, compared to non-CNN-based approaches. However, for pixel-level labeling, it is essential to develop models to capture contextual information (Chen et al., 2017). Image and feature pyramid methods use multi-level input images or feature maps. The proposed algorithm uses pyramid pooling (Zhao et al., 2017) to extract features from the input images. Methods such as region growth (Dias & Medeiros et al., 2018) have been proposed recently to create high-resolution segmentation refinement. However, such models cannot process high-resolution images due to computational limitations. Supervised and unsupervised encoder–decoder models have recently gained attention for segmentation tasks (Badrinarayanan et al., 2017; Chen et al., 2018; Rafiei & Adeli et al., 2018). The encoder–decoder model captures high-level semantics by reducing the spatial dimension using an encoding module and reconstructs the inputs using a decoder. Due to computational constraints for training semantic segmentation models, higher strides (Chen et al., 2017) are used, which leads to lower accuracies.

Transformer models (Vaswani et al., 2017) have gained popularity due to their performance on various sequence-based tasks, such as large language models. Such breakthroughs in transformer-based models for natural language processing sparked attention in the computer vision domain to perform vision tasks. As a result, transformer-based models have been developed for classification (Dosovitskiy et al., 2020), object detection (Carion et al., 2020), and segmentation (Ye et al., 2019).

This study introduces a new framework, Refined-Segment Anything Model (R-SAM), designed for high-resolution image segmentation. R-SAM utilizes the SAM (Lin et al., 2017) model as the base segmentation module to generate initial segmentation masks in a zero-shot way. An object detection model, the end-to-end object detection with transformers (Carion et al., 2020), is trained for labeling the segmented regions. A novel mask refinement module is developed and trained to improve and label any masks generated by SAM.

2 | VISION TRANSFORMERS (ViTs)

The utilization of transformers in the computer vision domain has the potential to bridge the gap between language processing and visual reasoning. The foundation of transformer models is the self-attention mechanism trained to comprehend the interdependencies among sequences. From this point of view, transformers are similar to recurrent neural networks (RNNs) (Hochreiter & Schmidhuber et al., 1997). RNNs can capture short-term context using the recursive procedure. However, transformers can learn long-range relationships between sequences through the attention mechanism. Though such a mechanism has been used in RNNs (Chaudhari et al., 2021; Correia & Colombini et al., 2021), transformers rely on the unique implementation of a multi-head attention mechanism that allows faster and parallel computation; this will enable transformers to be scalable to complex models and outperform when dealing with larger datasets (Khan et al., 2021; LeCun et al., 2015).

The concept of ViT was initially presented by Dosovitskiy et al. (2020). It has the potential to replace standard CNNs for large datasets. ViT was based on the original Transformer (Vaswani et al., 2017), where sequences of image patches were used instead of directly processing the pixel values of the entire image. It is worth emphasizing that their model was pre-trained and encoded prior knowledge about the images using the JFT-300 M dataset (Sun et al., 2017), which contains 300 million images. The data-efficient image transformers (DeiT; Touvron et al., 2021) model demonstrated that ViTs can be trained faster using mid-range datasets while maintaining high accuracy. The distillation approach in DeiT uses CNNs (teacher) to a transformer model (student). The standard ViT models had a fixed number of tokens through the network, which was a limitation for capturing spatial information. Transformers are built on two core components: self-attention and pre-training. Utilizing self-attention enables transformer models to capture long-range dependencies among sequences of features effectively, a challenge that traditional RNNs struggle to overcome. Meanwhile, pre-training involves training a network by leveraging a large, labeled or unlabeled dataset and then fine-tuning it on the target dataset (Devlin et al., 2018; Y. Liu et al., 2019). The self-attention mechanism aims to capture the relationships among all n entities within a sequence $X \in \mathbb{R}^{n \times d}$ by encoding them in terms of global information. In other

14678667, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/mice.13211 by University Of British Columbia, Wiley Online Library on [21/04/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

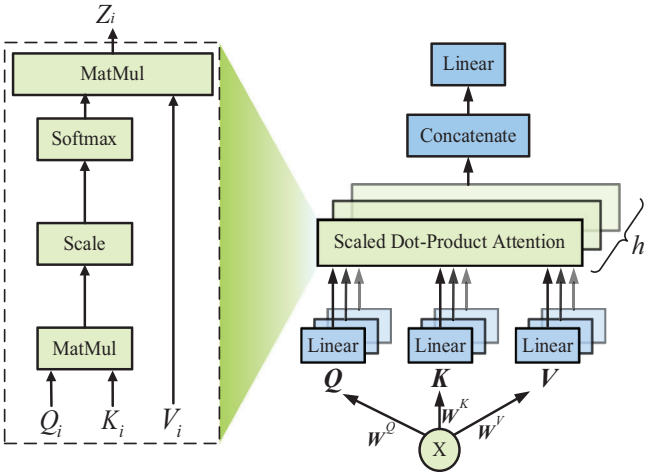

FIGURE 1 Parallelized attention mechanism (adapted from Vaswani et al., 2017).

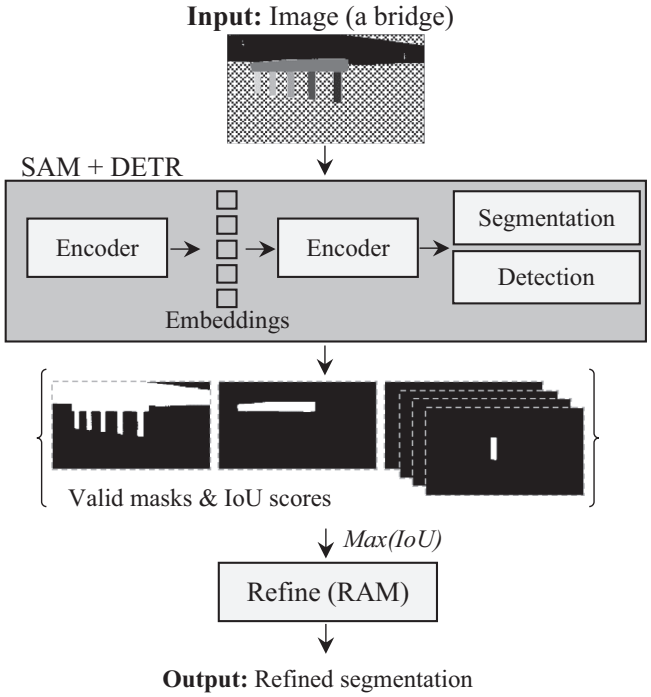

FIGURE 2 The overview of the Refined-Segment Anything Model (R-SAM). DETR, DEtection Transformer; IoU, intersection-over-union; RAM, Refine Anything Model.

words, it estimates the interactions between the different elements in the sequence, such as identifying which types of damage tend to co-occur in a given image of a structure. The embedding dimension d specifies the size of the vector space in which the sequence is represented, allowing for more complex and informative representations of the data. To accomplish this, the input sequence $\boldsymbol{X} = (x_1, x_2, \ldots, x_n)$ is projected onto a triplet of learnable matrices (i.e., weights), Queries $\boldsymbol{W}_Q = \in \mathbb{R}_{\mathrm{d}_{model} \times d_k}$, Keys $\boldsymbol{W}_K \in \mathbb{R}_{d_{model} \times d_k}$, and Values $\boldsymbol{W}_V \in \mathbb{R}_{d \times d_v}$. These matrices are then fed into a normalized dot-product attention mechanism as described by H. Zhang et al. (2019):

$$\boldsymbol{Z} = \mathrm{Softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_k}}\right)\boldsymbol{V} \tag{1}$$

where $\boldsymbol{Q} = \boldsymbol{X}\boldsymbol{W}_{\mathrm{Q}}$, $\boldsymbol{K} = \boldsymbol{X}\boldsymbol{W}_{\mathrm{K}}$, $\boldsymbol{V} = \boldsymbol{X}\boldsymbol{W}_{\mathrm{V}}$, and $\boldsymbol{Z} \in \mathbb{R}_{n \times d_v}$.

A multi-head self-attention mechanism (Figure 1) utilizes h parallel attention layers, or heads, to project the input sequence $\boldsymbol{X}$ onto representation subspaces, each with its learnable query, key, and value matrices $\{\boldsymbol{W}_{Q,i}, \boldsymbol{W}_{K,i}, \boldsymbol{W}_{V,i}\}_{i=1}^{h}$ (e.g., $h = 8$).

This approach is used to overcome the limitations of the standard attention mechanism in capturing multiple relationships simultaneously. To elaborate further, when a particular input is considered:

$$\boldsymbol{Z} = \boldsymbol{concat}\,(\boldsymbol{Z}_0, \boldsymbol{Z}_1, \ldots \boldsymbol{Z}_{\boldsymbol{h}-1})\,\boldsymbol{W}_{\boldsymbol{o}} \tag{2}$$

where $\boldsymbol{concat}(.)$ operation denotes concatenation, $\boldsymbol{W}_{\boldsymbol{o}}$ is the additional projection weight, and $\boldsymbol{W}_{\boldsymbol{o}} \in \mathbb{R}_{hd_v \times d_{model}}$ (Khan et al., 2021). The "linear" block in the figure refers to a linear transformation applied to project the feature map into different subspaces (Vaswani et al., 2017). The term "MatMul" refers to matrix multiplication, and before the softmax function, the attention values are scaled by the square root of the total number of keys ($1/\sqrt{d_k}$).

3 | PROPOSED METHOD

3.1 | Overview and abbreviations

Figure 2 provides a visual representation and an overview of the proposed pipeline, which consists of three main modules: SAM, Refine Anything Model (RAM), and DEtection Transformer (DETR). The SAM is used for zero-shot segmentation, the RAM is used to improve the segmentation, and DETR is used for labeling. Each module is explained in detail in the following sections. To segment all the components, whether structural or non-structural, depending on color and texture, the input red-green-blue (RGB) images are fed into the SAM module (Lin et al., 2017).

The RAM module is used to enhance the performance of SAM by refining the edges of the generated masks. The refined segmentation masks are finally labeled using the DETR, an object detection model based on transformers. The pipeline output is an instance segmentation, where every structural component or damage is detected and segmented. Each of the abovementioned components is explained in the following sections.

14678667, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/mice.13211 by University Of British Columbia, Wiley Online Library on [21/04/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

3.2 | SAM

In this study, SAM (Lin et al., 2017) is used to generate the initial segmentations for R-SAM. The rationale behind this selection is SAM's ability to generalize across novel tasks and domains without requiring custom data annotations of tuning (i.e., real-time zero-shot transfer). The original SAM has three main modules: the image and prompt encoder modules, which take input RGB images, prompts, output encoded embeddings, and the mask generator. For scalability, a pre-trained ViT adapts the model for high-resolution images. The image encoder's output is fed into the mask decoder head, which outputs the final masks. While the primary intent of the original SAM was to be used for prompt-based segmentation (Lin et al., 2017), accepting input masks, points, and text prompts, the text prompting of SAM has not yet been released in the original version. This gap motivated the development of Grounded-Segment-Anything (Grounded-SAM; Ren et al., 2024), which utilizes both SAM and Grounding DINO (S. Liu et al., 2023) models for the identification of segmented objects using text inputs. However, these models cannot detect types of structural components and damages. The pre-trained SAM is implemented in this study, and additional information, including the pre-trained models, can be found in the original paper (Lin et al., 2017).

3.3 | RAM

This section explains the proposed RAM module of R-SAM, inspired by the models using cascade features, in detail (Ali et al., 2018; H. K. Cheng et al., 2020). This research uses a stack of multiple RAM modules to achieve high-resolution segmentation results.

The refinement module is based on a pyramid scene parsing network (PSP-Net) (H. K. Cheng et al., 2020; Zhao et al., 2017) architecture with three key components: (1) the ResNet34 as the backend pre-trained model for feature extraction, (2) the pyramid pooling module (PPM) as the mask decoder, and (3) the upsampling module. The RAM module receives a batch of input RGB images and the multi-resolution segmentation masks (i.e., with strides of 1, 2, 4, and 8) and passes them through the backend (He et al., 2016) to extract the feature map. Then, the acquired feature map is passed through the PPM to perform adaptive average pooling at varying scales, resulting in an output that captures a representation of the input at multiple scales (Zhao et al., 2017). A 1×1 convolution layer, with rectified linear unit (ReLU) activation (Nair & Hinton et al., 2010), takes the concatenated outcomes to decrease the depth of the feature map (H. K. Cheng et al., 2020).

Using the feature maps derived from the preceding layer, the upsampling module performs its operations by leveraging bilinear interpolation and incorporating concatenation with the feature maps from the skip connections from the backend. The obtained feature maps are passed through a sequence of convolution layers, with a sigmoid activation function, to generate the segmentation mask. The outputs are upscaled to generate the masks for the next iteration. The skip connections are incorporated into the design to let the network learn mask features at the pixel level by retaining the lost information from different steps. The multi-resolution segmentation mask generation helps fix the imperfect segmentation masks' overall structure and boundary (H. K. Cheng et al., 2020).

The cross-entropy (CE) loss function is employed for the coarser stride 8 output to detect the global features. However, for finer stride 1, the L1+L2 loss functions (i.e., mean absolute error and mean squared error) push the model to pay attention to the local pixel-level features. The PyTorch library provides these loss functions (Paszke et al., 2017).

$$\mathcal{L}_8 = \mathcal{L}_{CE} \tag{3}$$

$$\mathcal{L}_1 = \mathcal{L}_{L1+L2} + \beta \mathcal{L}_{grad} \tag{4}$$

$$\mathcal{L}_2 = \mathcal{L}_4 = \frac{1}{2}\left(\mathcal{L}_{L1+L2} + \mathcal{L}_{CE}\right) \tag{5}$$

where $\mathcal{L}_{grad}$ is the gradient loss that is used for the finer stride 1, which is defined as follows (H. K. Cheng et al., 2020):

$$\mathcal{L}_{grad} = \frac{1}{n}\sum_i \left\| \nabla\left(f_m\left(x_i\right)\right) - \nabla\left(f_m\left(y_i\right)\right) \right\|_1 \tag{6}$$

where ∇ denotes the gradient or edge detector operator, and $f_m(.)$ is the mean filter (Kanopoulos et al., 1988). The overall loss function is defined as $\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_4 + \mathcal{L}_8$. Multiple RAM modules are recursively used to implement the trained model for high-resolution segmentation of imperfect input segmentation and to capture pixel-level details of structural components or damages accurately. Each iteration replaces one output mask as illustrated in Figure 3. In this study, the RAM module consists of five iterations that could be reduced depending on the complexity of the problem. As a general recommendation, smaller objects require deeper networks with more than three iterations, while larger components can be refined using fewer iterations. In other words, during the first three iterations, where input masks are passed to the subsequent iterations, the model is forced to learn the global refinement of larger objects. The input of the RAM module is the color image and the initial

14678667, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/mice.13211 by University Of British Columbia, Wiley Online Library on [21/04/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

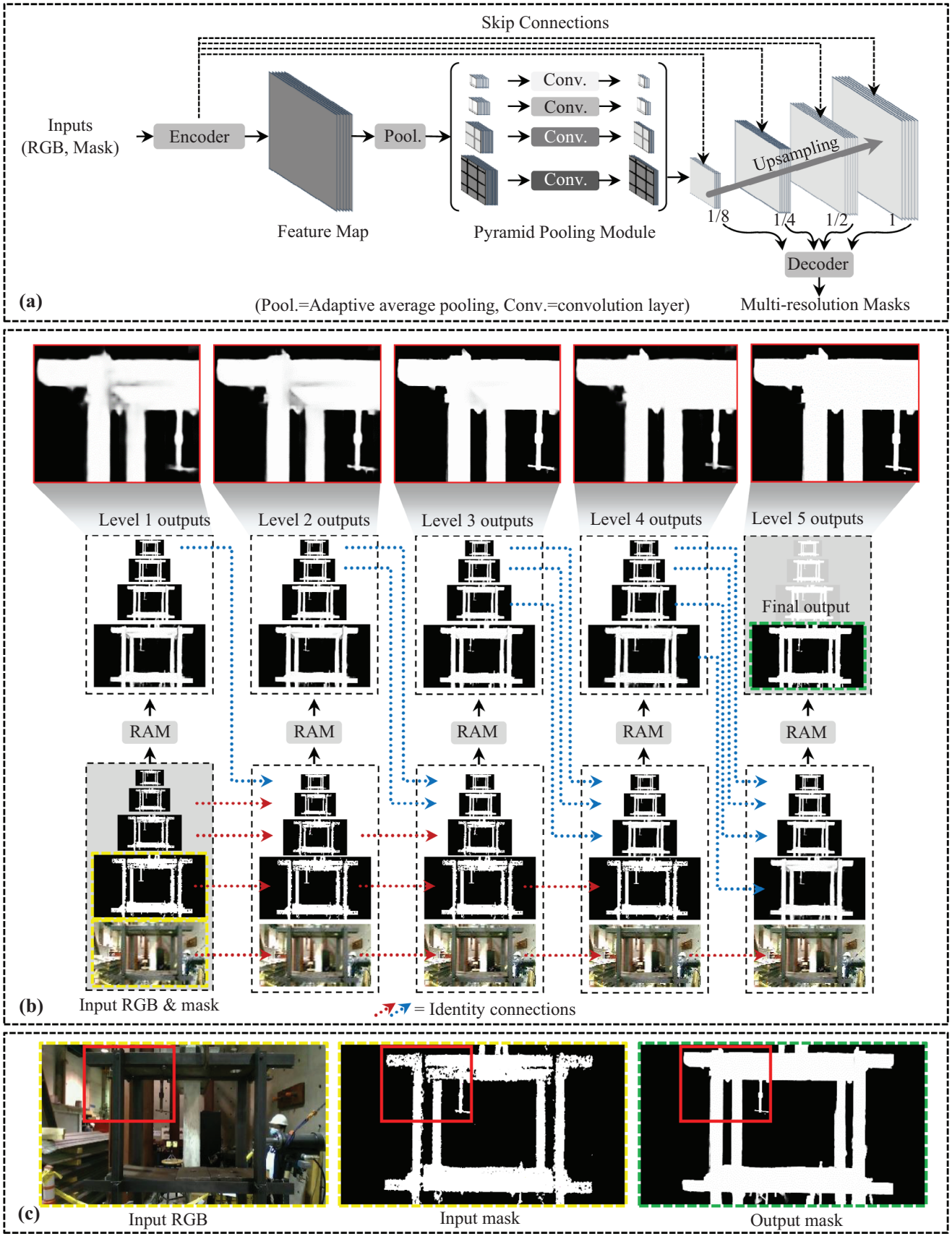

FIGURE 3 Implementation pipeline of the Refine Anything Model (RAM): (a) RAM, (b) multi-level refinement, and (c) inputs and output. RGB, red-green-blue.

14678667, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/mice.13211 by University Of British Columbia, Wiley Online Library on [21/04/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

mask. For the first level of refinement, the algorithm generates multiscale masks before feeding them through the RAM using interpolation. Identity connections are used to pass the inputs and outputs for subsequent iterations.

3.4 | DETR

This section provides an elaborate explanation of the detection module of R-SAM, which is inspired by the research projects that used the advantages of transformers in computer vision tasks (Carion et al., 2020; Dosovitskiy et al., 2020; Khan et al., 2021). Transformers are used for end-to-end detection tasks in three ways: CNN backbone with a transformer head, transformer backbone with an R-CNN head, and pure transformer. DETR, the end-to-end object detection with transformers (Carion et al., 2020), uses the first approach to predict bounding boxes around objects. This way, it is possible to predict multiple objects and their relationships from a single shot by performing bipartite matching.

One key feature of DETR is that it does not have hand-designed components to encode prior knowledge, such as proposal box coordinates (Carion et al., 2020). This allows DETR to perform object detection tasks without knowledge of complex detection tasks. The backbone of the DETR model creates spatial feature maps, where every input image is first zero-padded to match the shape of the largest image in the dataset. A convolution layer is utilized to decrease the channel dimension of the input spatial feature map into a one-dimensional sequence of features. The encoder and decoder modules of the transformer incorporate multi-head attention as well as a feed-forward network. Unlike the previous transformers (Vaswani et al., 2017), DETR can decode multiple objects in parallel. Furthermore, positional encoding is used to embed information regarding the position of the elements within input sequences. The key reasons for incorporating positional encoding are: (a) transformers, unlike RNNs, are permutation-invariant, and they cannot understand the sequential order of the inputs inherently, and it helps to consider the order of elements; (b) to capture spatial relationships, such as proximity, distance, and orientation, which is crucial for object detection; and (c) without using positional encoding, a transformer-based model may treat two similar inputs (i.e., symmetry) at the different locations as equivalent. The DETR generates predictions per iteration and achieves optimal bipartite matching by associating the predicted bounding boxes with the corresponding ground truth bounding boxes. Further details regarding the DETR implementation can be found in the original paper (Carion et al., 2020).

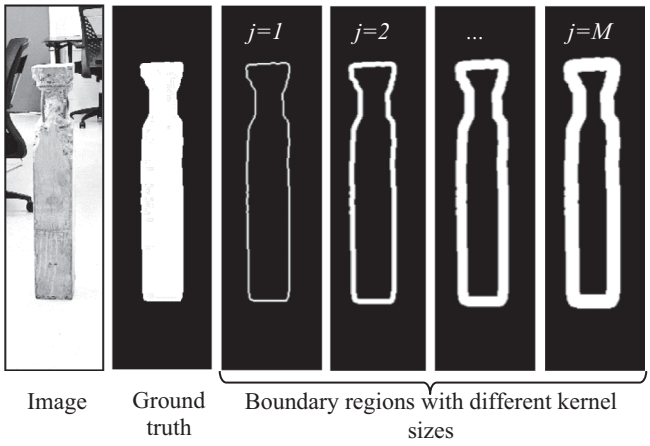

FIGURE 4 The ground truth mask of a reinforced concrete column and the corresponding boundary regions with different radiuses.

3.5 | Evaluation metrics

Two accuracy metrics were employed to evaluate the performance of the segmentation models: mean intersection-over-union (mIoU) score across all classes and mean boundary accuracy (mBA). The mIoU is calculated by dividing the pixel intersection by the pixel union,

$$mIoU = \frac{1}{n_c} \sum_{i=1}^{n_c} \frac{TP_i}{TP_i + FP_i + FN_i} \tag{7}$$

where n_c is the number of classes, TP_i, FP_i, FN_i are true positive, false positive, and false negative for class i, respectively. To calculate the BA of the refined masks, the binary map of boundary regions for the ground truth masks was generated using the morphological gradient method implemented in the OpenCV library (Bradski & Kaehler et al., 2008; OpenCV et al., 2015). Different disk-shaped structuring elements of a varying radius within the range $[1,\ \max(height, width)/200]$ (Figure 4).

The mean bounding accuracy is calculated as the mean of the ratio of the number of accurately identified boundary pixels to the sum of boundary pixels.

$$mBA = \frac{1}{M} \sum_{j=1}^{m} \frac{1}{N_j} \sum_{i=1}^{n} I_{i,j} \tag{8}$$

where mBA represents the BA for a mask, $I_{i,j}$ is the ith inbound pixel (correctly predicted boundary pixel) of the segmentation mask for the jth boundary map. N and M are the total numbers of boundary pixels of the ground truth mask and the total number of boundary masks that were generated ($M = 5$ in this study), respectively.

14678667, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/mice.13211 by University Of British Columbia, Wiley Online Library on [21/04/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

4 | TRAINING AND RESULTS

All three modules of the proposed R-SAM were developed or implemented utilizing the PyTorch framework. Details about the training of the refinement and object detection models are provided below in this section.

4.1 | Training refinement model

The RAM module was trained using a diverse segmentation dataset that includes CrackForest (Y. Shi et al., 2016) with 329 images, concrete crack segmentation dataset (Özgenel et al., 2019) with 458 high-resolution images, structural material semantic segmentation dataset (Bianchi & Hebdon et al., 2021) with 3817 images, and extended complex scene semantic description dataset (ECSSD) (J. Shi et al., 2015) with 1000 images, MSRA salient object database (M. -M. Cheng et al., 2014) with 10,000 images, FSS-1000 (Li et al., 2020) with 10,000 images, and DUT-OMRON pixel-wise dataset (Yang et al., 2013) with 15,000 images. To ensure that the RAM module can be applied to various scenarios, existing pre-trained models, such as SAM, were not used to generate additional segmentation masks.

Data augmentation techniques, including image flipping and random brightness, were considered in this study to add variety. All the pixel values were normalized with $\mu = [0.485, 0.456, 0.406]$ and $\sigma = [0.229, 0.224, 0.225]$ for image data, and $\mu = [0.5]$ and $\sigma = [0.5]$ for segmentation masks. The image and mask data were converted into PyTorch tensors before feeding into the models. Furthermore, the RAM module was trained using perturbed ground truth, as suggested by H. K. Cheng et al. (2020), to improve the robustness of the R-SAM. Adam optimizer (Kingma & Ba et al., 2014) was used with 10,000 iterations with a batch size of 8. Figure 3 shows the segmentation improvement using the five-level refinement.

4.2 | Training object detection model

This study trained the detection model to generate the bounding boxes for each object detection task (e.g., crack detection, component detection, and damage detection). The image dataset for structural components and damage detection of concrete columns was created using experimental reinforced column tests at the Structural Laboratory at the University of British Columbia (UBC). The images were taken in different locations and lighting conditions, and additional augmentation techniques such as random crops, resizing, rotating, blurring, and added noise during the training were used to generalize the trained model. A total of 250 images, with 4032 × 3024 resolution, were annotated with bounding boxes around the concrete column and the damaged regions. The accurate segmentation annotation dataset was created using the green screen technique; this dataset was used to validate the effectiveness of the proposed method since the majority of the openly available datasets are annotated using manual segmentation techniques and are not highly accurate. The dataset was split into training, validation, and testing subsets with 70:15:15 ratios. The training was done with the optimized key hyperparameters: learning rate = 0.00005, batch size = 8, weight decay = 0.0001, epochs = 50, backbone = ResNet34, position embedding = sine, encoder layers = 6, decoder layers = 6, hidden layer dimension = 256, dropout = 0.2, number of queries = 5. The test results are shown in Figure 5a for a batch of images from the test subset. This figure shows that the model can effectively detect the object of interest (i.e., component) and the regions of interest for damage detection. These bounding boxes are used in R-SAM to identify the segmentation region accurately.

5 | DISCUSSIONS

5.1 | Comparative analysis of segmentation models

A total of 27 segmentation models were trained with different configurations to investigate performance improvement using the RAM module. These models are based on the U-Net (Ronneberger et al., 2015), U-Net++ (Z. Zhou et al., 2018), MA-Net (Fan et al., 2020), LinkNet (Chaurasia & Culurciello et al., 2017), feature pyramid network (FPN) (Kirillov et al., 2019, Lin et al., 2017), and DeepLabV3+ (Chen et al., 2018), and with different encoders including EfficientNet (Tan & Le et al., 2019), ResNet (He et al., 2016), MobileNet (Howard et al., 2017; Sandler et al., 2018), MiT (Touvron et al., 2019), and VGG (Simonyan & Zisserman et al., 2014). The results are presented in Table 1 for *mIoU* and *mBA* metrics. In addition, the results were compared with the same models refined using the proposed RAM module for achieving more accurate segmentation results. The proposed R-SAM model achieved a mIoU of 97.23% with a mBA of 87.06%, which exhibits superior performance, compared to the other models studied in this research. Figure 5b,c shows the visual comparison between different segmentation models and compares them with the results obtained from SAM (using DETR for detection) and R-SAM. Except for the U-Net and U-Net++ with EfficientNet-b1 encoders, the rest of the model achieved a high IoU after refinement. However, R-SAM has the highest BA due to the initial segmentation results with a higher IoU and BA. The FPN model based on ResNet18

14678667, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/mice.13211 by University Of British Columbia, Wiley Online Library on [21/04/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

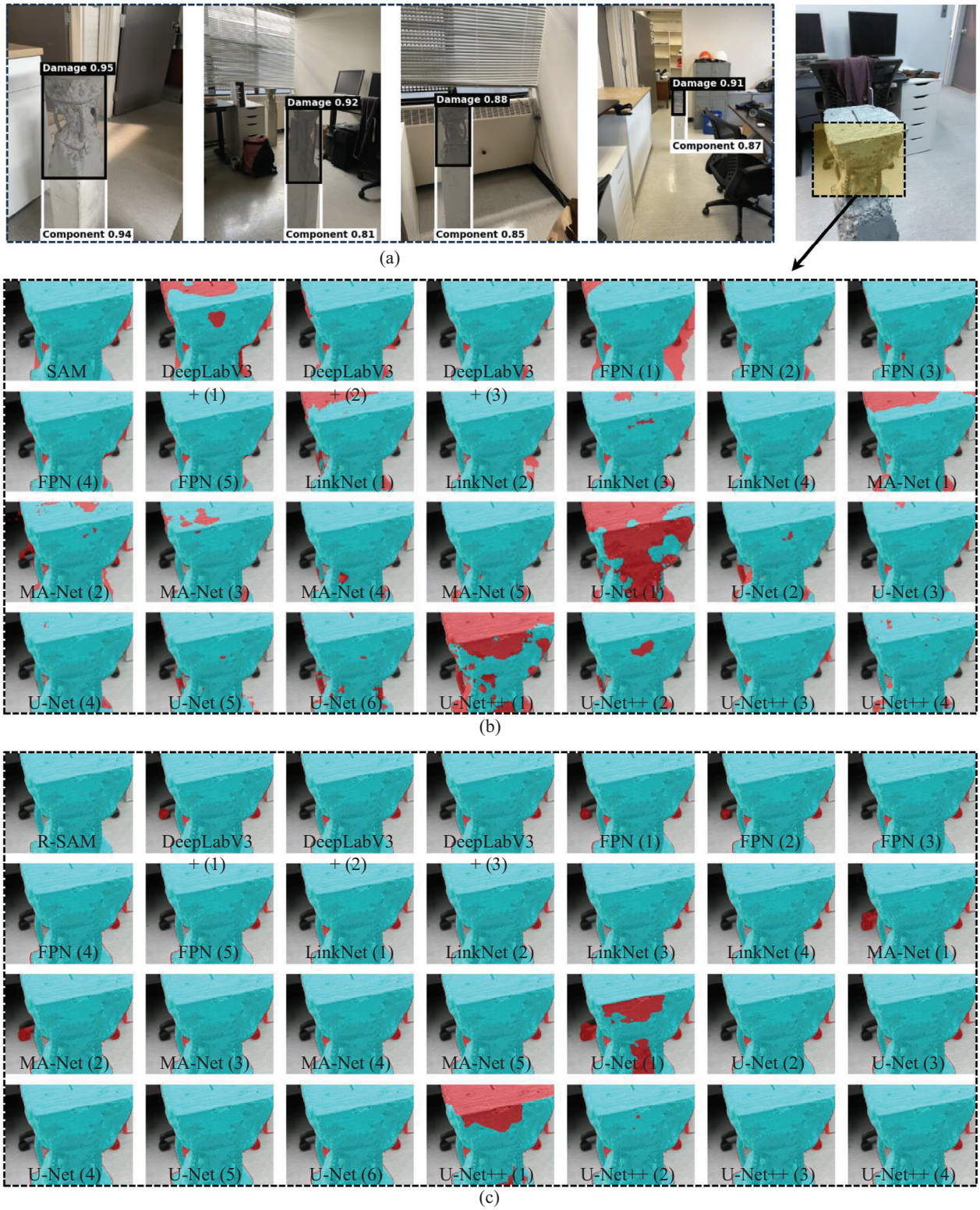

FIGURE 5 Visual comparison of different segmentation methods: (a) detection results and confidence scores, (b) segmentation results before applying refinement, and (c) segmentation results after applying refinement (true and false results are shown in cyan and red colors, respectively).

14678667, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/mice.13211 by University Of British Columbia, Wiley Online Library on [21/04/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 1 Comparative analysis of various semantic segmentation models with and without refinement.

Model	Encoder	$mIoU$ (%)	$mIoU_{RAM}$(%)	Δ_{IoU} (%)	mBA (%)	mBA_{RAM} (%)	Δ_{BA} (%)	mIT (s)
DeepLabV3+ (1)	EfficientNet-b0	76.10	92.25	16.15	65.61	80.21	14.61	1.54
DeepLabV3+ (2)	ResNet34	90.92	96.58	5.66	78.27	81.12	2.84	1.92
DeepLabV3+ (3)	ResNet50	88.75	96.18	7.43	76.89	81.04	4.15	2.62
FPN (1)	EfficientNet-b0	68.14	94.83	26.69	57.35	80.63	23.28	1.57
FPN (2)	MobileNetV2	91.64	96.81	5.17	75.80	81.27	5.47	1.04
FPN (3)	ResNet18	**94.29**	96.84	2.56	77.25	81.30	4.05	1.26
FPN (4)	ResNet34	91.52	96.84	5.32	72.29	81.24	8.95	1.68
FPN (5)	ResNet50	89.78	96.37	6.58	72.14	80.86	8.73	2.21
LinkNet (1)	EfficientNet-b0	71.84	77.95	6.11	71.14	74.64	3.50	1.23
LinkNet (2)	MobileNetV2	90.73	96.65	5.92	80.71	81.13	0.42	0.83
LinkNet (3)	ResNet18	91.63	96.64	5.00	81.55	81.00	−0.55	1.01
LinkNet (4)	ResNet34	92.68	96.79	4.11	83.85	83.26	−0.59	1.60
MA-Net (1)	EfficientNet-b0	78.78	91.51	12.73	76.20	79.95	3.75	1.84
MA-Net (2)	MiT-b0	81.67	96.39	14.72	73.75	81.11	7.36	2.11
MA-Net (3)	MobileNetV2	92.36	96.52	4.16	84.12	82.98	−1.14	1.58
MA-Net (4)	ResNet18	92.56	96.72	4.16	**84.81**	82.75	−2.07	1.63
MA-Net (5)	ResNet34	88.82	95.75	6.93	79.06	81.17	2.11	2.08
U-Net (1)	EfficientNet-b1	55.21	70.11	14.90	68.88	74.62	5.74	1.29
U-Net (2)	MobileNetV2	86.17	96.83	10.66	82.16	81.25	−0.91	0.80
U-Net (3)	ResNet34	90.86	96.83	5.97	83.11	82.28	−0.83	1.15
U-Net (4)	ResNet50	86.50	95.39	8.89	82.95	82.74	−0.21	1.83
U-Net (5)	VGG16	85.17	96.63	11.47	78.51	81.22	2.72	4.97
U-Net (6)	VGG19	85.28	96.81	11.53	78.99	81.31	2.31	5.53
U-Net++ (1)	EfficientNet-b1	56.64	66.29	9.65	69.48	74.31	4.83	1.65
U-Net++ (2)	MobileNetV2	86.41	96.83	10.41	83.40	83.26	−0.15	0.93
U-Net++ (3)	ResNet34	81.44	96.84	15.41	77.82	81.26	3.44	1.94
U-Net++ (4)	ResNet50	87.41	95.05	7.64	83.35	81.92	−1.43	3.28
R-SAM	–	91.24	**97.23**	60.01	79.61	**87.06**	7.45	0.86

Maximum values are indicated in bold font.

Abbreviations: mBA, mean boundary accuracy; mIOU, mean intersection-over-union; mIT, mean inference time; RAM, Refine Anything Model; R-SAM, Refined-Segment Anything Model.

achieved a high IoU. However, with lower BA, compared to SAM+DETR (R-SAM before refinement), the final refined mask accuracy is not high. These results highlight the importance of considering the BA in addition to the IoU. Considering the zero-shot segmentation and refinement, using SAM and RAM modules, the proposed framework has promising potential for real-world applications where segmentation data may not be available, and rapid assessments are required. Further analysis of the mean inference time (mIT) for each model was done using a portable computer with a RTX-3080Ti graphics processing unit.

6 | CASE STUDIES

Three applications for R-SAM are presented in this section, highlighting its versatility and effectiveness in various aspects of SHM.

6.1 | Accurate crack segmentation

Compared to low-resolution crack images, high-resolution images of cracks provide detailed information for accurately evaluating structural conditions (Chu & Chun et al., 2024). One of the promising applications of the proposed method is refining low-resolution segmentation masks. Such masks could be generated using lightweight models such as Mask R-CNN (He et al., 2017). Figure 6a shows the performance of the trained RAM module for concrete crack segmentation, where the mask improvement is applied to the upsampled output of a pre-trained Mask R-CNN. This application indicates the proposed RAM module's potential in segmenting high-resolution images while using pre-trained light models that output low-resolution masks.

14678667, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/mice.13211 by University Of British Columbia, Wiley Online Library on [21/04/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

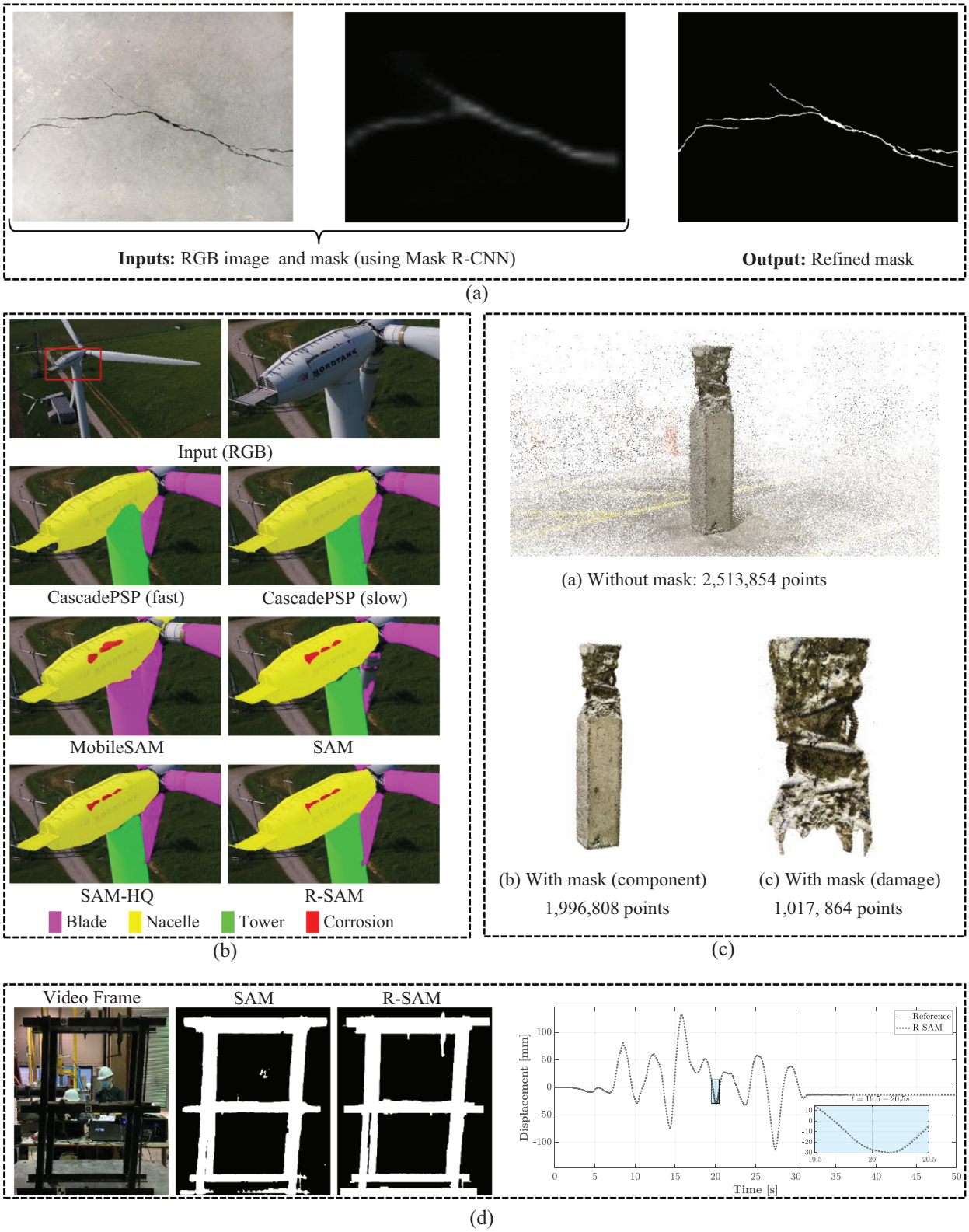

FIGURE 6 Applications of R-SAM in structural health monitoring: (a) crack segmentation refinement, (b) multiclass segmentation and damage detection of wind turbine using different SAM-based methods, (c) 3D reconstruction using original images and multi-level masked images, and (d) real-time segmentation and tracking. CNN, convolution neural network; RGB, red-green-blue.

14678667, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/mice.13211 by University Of British Columbia, Wiley Online Library on [21/04/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

6.2 | Multiclass segmentation

To demonstrate the effectiveness of R-SAM in the accurate segmentation of multiple classes, the publicly available DTU–Drone inspection images of the wind turbine (Shihavuddin & Chen et al., 2018) were used to train the DETR model. For this purpose, the dataset (5280 × 2970 images) was annotated using a multiclass annotation approach, where each instance was labeled with a bounding box. The results of this case study are presented in Figure 6b, which shows that R-SAM can identify and segment small and large components such as blades, nacelles, towers, and potential damages such as corrosion. The efficiency of the refinement module is compared with the previous refinement model CascadePSP (H. K. Cheng et al., 2020), CascadePSP-Fast with lower accuracy, and CascadePSP-Slow with higher accuracy. As the results show, both models face challenges in refining small objects, which the current study addresses using the RAM module.

Furthermore, compared to the other SAM-based models (i.e., SAM, Mobile-SAM; C. Zhang et al., 2023), and HQ-SAM (Ke et al., 2023) model, R-SAM demonstrates a significant improvement in refining the boundaries of each segmented region, which significantly enhances the overall performance and capabilities of R-SAM for SHM applications. This capability holds significant value as it enables timely maintenance interventions and effective mitigation of potential risks, ensuring the efficient operation of wind energy systems and maximizing their lifespan. By providing precise and reliable information about the condition of wind turbines, R-SAM empowers operators and maintenance teams to proactively address issues, optimize performance, and safeguard the long-term sustainability of these critical energy assets.

6.3 | 3D reconstruction using masked images

Structural from motion (SfM) techniques have revolutionized photogrammetry by providing a cost-effective method for 3D reconstruction using RGB images (Westoby et al., 2012). Despite the effectiveness of SfM methods, inaccuracies in the generated 3D models may occur due to the presence of reflections, lighting, occlusions, and repeated patterns in the scene (Pan & Yang et al., 2023). One approach to address these issues and improve the accuracy of SfM is using masked images. Masking can remove unwanted objects, such as moving people or robots, which may cause artifacts in the 3D model. Limiting the feature extraction to specific regions of interest allows a faster and more accurate 3D reconstruction using high-resolution images. This is particularly important when using limited computational power as it reduces computation costs and data storage requirements.

As an ablation study, the application of the proposed R-SAM in the 3D reconstruction of structural components using masked images was investigated. For this purpose, COLMAP (Schonberger & Frahm et al., 2016) was used. The results for multi-level segmentation (i.e., component-level and damage-level) masks were generated automatically, and the background pixels were removed from the computation, leaving power for more accurate and detailed 3D reconstruction of the objects of interest. The results of this experiment are shown in Figure 6c. The results demonstrate that the proposed R-SAM method can produce finer and denser point clouds of damaged regions while requiring fewer points and achieving faster results than the original images.

6.4 | Marker-free segmentation and tracking

Accurate measurement of structural vibrations plays a critical role in SHM. Over the past few years, a growing interest has been in employing cost-effective computer vision approaches to measure structural vibrations (Azimi et al., 2020). Optical flow-based algorithms have garnered significant attention among the various computer vision algorithms explored for motion detection. One widely used optical flow method is Kanade–Lucas–Tomasi (KLT; Tomasi & Kanade et al., 1991), which has demonstrated high precision in motion tracking, particularly in a controlled environment. However, despite the efficacy of the KLT technique, it is sensitive and less robust to light change. Although recent studies utilized DL for marker detection for structural vibration measurement, areas can still be improved to enhance their effectiveness and reliability (Pan et al., 2023).

A case study was conducted to exhibit the capability of R-SAM in accurate segmentation and marker-free tracking of structural components. The recorded video, recorded in 3840 × 2160 resolution at 60 frames per second, was provided by Smart Structures at the UBC. During this experiment, a light panel was used to change the light intensity during the test to investigate the sensitivity and robustness of the model. The results of this case study are shown in Figure 6d. R-SAM can significantly refine the boundaries of the regions of interest (full-frame in this example), which can be used for accurate measurement.

R-SAM provides a marker-free approach to tracking the entire frame using a single binary mask, which can be generalized to track multiple components simultaneously. This approach allows extending the applications of R-SAM to model updating and system identification.

14678667, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/mice.13211 by University Of British Columbia, Wiley Online Library on [21/04/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 2 Comparative analysis of the proposed method for tracking applications.

Horizontal acceleration name	Earthquake name	Level	Maximum displacement (mm)		RMSE	R^2
			Measured	R-SAM		
RSN718_SUPER.A_A-IVW090	Superstition Hills, 1987	Base	70.05	71.19	1.62	0.99
		Story1	88.31	89.80	1.68	0.99
		Story2	132.49	135.21	2.05	0.98
RSN52_SFERN_AZP045	Superstition Hills, 1987	Base	70.10	71.72	2.31	0.99
		Story1	115.22	117.12	1.64	0.99
		Story2	179.67	183.29	2.01	0.99
RSN52_SFERN_AZP045	San Fernando, 1971	Base	54.46	54.84	0.69	0.99
		Story1	110.73	128.72	16.24	0.94
		Story2	138.68	146.19	5.40	0.99

Abbreviations: R^2, coefficient of determination; RMSE, oot mean square errors; R-SAM, Refined-Segment Anything Model.

Table 2 summarizes the results of experimental studies for two seismic events, namely, the Superstition Hills 1987 and San Fernando 1971 earthquakes. The maximum displacements, root mean square errors, and coefficient of determination (R^2) are detailed for the R-SAM results, compared to the measured displacements at different floor levels. The comparison highlights a close agreement between the reference and the obtained results. It is worth noting that the above results were obtained using stationary video recording cameras, where no occlusion occurs. In the case of occlusion (e.g., partially visible objects), advanced transformer-based tracking models such as co-tracking (Karaev et al., 2023) can be used to track the partially visible or temporary invisible points. Therefore, the predicted tracking data points, in a zero-shot way, are fed into the R-SAM.

7 | LIMITATIONS AND FURTHER STUDIES

While the effectiveness of the proposed pipeline meets acceptable requirements for the conducted experiments and considering large objects in the input images, it is not perfect. The SAM module may not perfectly segment or detect small components or disconnected and occluded objects. In addition, the RAM module may require high GPU memory for very large images to generate high-resolution segmentation with higher accuracy.

One option to segment ultra-high resolution images without resizing the dimensions is to refine the segmentation using the cropped regions from the original images without allocating more GPU memory. Therefore, square $L \times L$ patches of the images are obtained at specific strides, and the obtained images are fed into the RAM module. The average of the output masks is used as the final refined mask to address the disagreement among the outputs from different overlapping patches. Figure 7 shows the performance of the RAM module when with varying values of L. Using a lower L produces segmentation with lower accuracies, while higher L results is better performance. For a typical GPU with 16 GB of memory, the maximum L value would be 2000 pixels. It has been noted that maintaining a minimum value of $L = 600$ is crucial to avoid any adverse effects on enhancing accuracy. Although there is a substantial increase in execution time for L values greater than 1600, the mBA does not change very much for larger values of L.

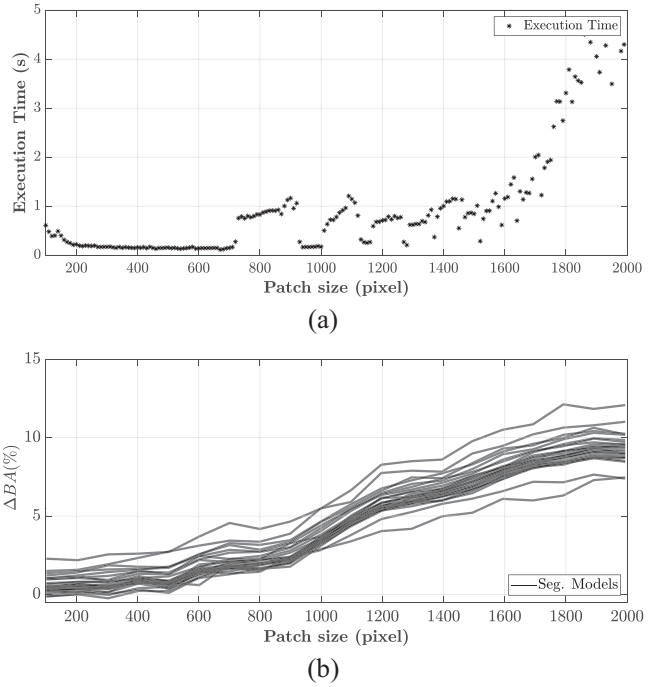

FIGURE 7 Effect of patch size on (a) the execution time of RAM module and (b) the boundary accuracy improvement (Δ_{BA} in Table 1).

14678667, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/mice.13211 by University Of British Columbia, Wiley Online Library on [21/04/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Future studies would optimize the proposed R-SAM for real-time applications using typical GPUs by segmenting and refining the detected regions of interest (i.e., bounding boxes), which may require performing the object detection before SAM. It is worth noting that the latest version of PyTorch allows reconfiguring and running the SAM model eight times faster than the original model (PyTorch et al., 2023).

Training a model from scratch may require a diverse dataset. However, transfer learning, regularization, and augmentation techniques may be considered to avoid overfitting issues associated with limited datasets. Future studies may also bridge the gap between vision-based and text-based models by developing object detection models using text prompts, such as Grounding DINO (S. Liu et al., 2023) and Grounded-SAM (Ren et al., 2024).

Last, it is essential to systematically investigate trained weights and address concerns related to the interpretability of models using various methods, such as gradient-weighted class activation mapping (i.e., GradCAM; Selvaraju et al., 2017) and attention visualization (Chefer et al., 2021; B. Zhou et al., 2016).

8 | CONCLUSION

In this study, a new framework, called R-SAM, is proposed to address the problem of automated damage assessment in civil infrastructure using high-resolution images. The proposed framework comprises three modules: detection, segmentation, and refinement. The detection module detects objects of interest from the input images, and the pre-trained SAM is utilized to generate the initial segmentation mask. A novel refinement model is trained to improve the accuracy through a multi-stage refinement process. The proposed method was implemented and trained using diverse datasets. The performance of the trained refinement module was evaluated by comparing it with other segmentation techniques. The findings indicate that R-SAM outperforms all the models with a mIoU of 97.23% and a mBA of 87.06%. The results show that R-SAM has promising potential for real-world applications where segmentation data may not be available, and rapid assessments are required. While the proposed pipeline demonstrates effectiveness for large objects in input images, limitations arise in accurately segmenting small or occluded components. Future research would focus on optimizing the R-SAM for real-time applications such as simultaneous tracking and segmentation.

ACKNOWLEDGMENTS

The authors would like to thank Armin Dadras Eslamlou and Mohammad-Ali Heravi for their scientific comments and support in developing the transformer models and colleagues Sina Tavasoli and Xiao Pan for providing the annotated dataset of concrete columns and shake-table experimental tests data. Partial support for this study was provided through utilizing computational resources offered by Advanced Research Computing (ARC) at the University of British Columbia.

REFERENCES

Ali, R., Gopal, D. L., & Cha, Y -J. (2018). Vision-based concrete crack detection technique using cascade features. In S. Hoon (Ed.), *Sensors and smart structures technologies for civil, mechanical, and aerospace systems 2018* (Vol. 10598). International Society for Optics and Photonics. https://www.spiedigitallibrary.org/conference-proceedings-of-spie/10598/2295962/Vision-based-concrete-crack-detection-technique-using-cascade-features/10.1117/12.2295962.short

Azimi, M., Eslamlou, A. D., & Pekcan, G. (2020). Data-driven structural health monitoring and damage detection through deep learning: State-of-the-art review. *Sensors*, *20*(10), 2778.

Azimi, M., & Pekcan, G. (2020). Structural health monitoring using extremely compressed data through deep learning. *Computer-Aided Civil and Infrastructure Engineering*, *35*(6), 597–614.

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.

Bianchi, E., & Hebdon, M. (2021). *Structural material semantic segmentation dataset*. [Data set]. University Libraries, Virginia Tech. https://data.lib.vt.edu/articles/dataset/Structural_Material_Semantic_Segmentation_Dataset/16624648

Bradski, G., & Kaehler, A. (2008). Learning OpenCV: computer vision with theOpenCV library. O'Reilly Media, Inc. https://www.oreilly.com/library/view/learning-opencv/9780596516130/

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-end object detection with transformers. *European Conference on Computer Vision*, Glasgow, UK (pp. 213–229).

Chaudhari, S., Mithal, V., Polatkan, G., & Ramanath, R. (2021). An attentive survey of attention models. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *12*(5), 1–32.

Chaurasia, A., & Culurciello, E. (2017). LinkNet: Exploiting encoder representations for efficient semantic segmentation. *2017 IEEE Visual Communications and Image Processing (VCIP)*, St. Petersburg, FL (pp. 1–4).

Chefer, H., Gur, S., & Wolf, L. (2021). Transformer interpretability beyond attention visualization. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN (pp. 782–791).

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (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.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. In V. Ferrari, M. Hebert, C. Sminchisescu, & Y. Weiss (Eds.), *European conference on computer*

14678667, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/mice.13211 by University Of British Columbia, Wiley Online Library on [21/04/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

vision (ECCV), (pp. 801–818). Springer. https://ieeexplore.ieee.org/document/7913730

Cheng, H. K., Chung, J., Tai, Y.-W., & Tang, C.-K. (2020). CascadePSP: Toward class-agnostic and very high-resolution segmentation via global and local refinement. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA (pp. 8890–8899).

Cheng, M.-M., Mitra, N. J., Huang, X., Torr, P. H., & Hu, S.-M. (2014). Global contrast based salient region detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*(3), 569–582.

Chu, H., & Chun, P.-J. (2024). Fine-grained crack segmentation for high-resolution images via a multiscale cascaded network. *Computer-Aided Civil and Infrastructure Engineering*, *39*(4), 575–594.

Correia, A. D. S., & Colombini, E. L. (2021). Attention, please! A survey of neural attention models in deep learning. arXiv preprint. arXiv:2103.16775. https://arxiv.org/abs/2103.16775

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint. arXiv:1810.04805. https://arxiv.org/abs/1810.04805

Dias, P. A., & Medeiros, H. (2018). Semantic segmentation refinement by Monte Carlo region growing of high confidence detections. In C. Jawahar, H. Li, G. Mori, & K. Schindler (Eds.), *Asian conference on computer vision* (pp. 131–146). Springer.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., & Gelly, S. (2020). An image is worth 16×16 words: Transformers for image recognition at scale. arXiv preprint. arXiv:2010.11929. https://arxiv.org/abs/2010.11929

Fan, T., Wang, G., Li, Y., & Wang, H. (2020). MA-Net: A multiscale attention network for liver and tumor segmentation. *IEEE Access*, *8*, 179656–179665.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy (pp. 2961–2969).

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV (pp. 770–778).

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint. arXiv:1704.04861. https://arxiv.org/abs/1704.04861

Kanopoulos, N., Vasanthavada, N., & Baker, R. L. (1988). Design of an image edge detection filter using the Sobel operator. *IEEE Journal of Solid-State Circuits*, *23*(2), 358–367.

Karaev, N., Rocco, I., Graham, B., Neverova, N., Vedaldi, A., & Rupprecht, C. (2023). CoTracker: It is better to track together. arXiv preprint. arXiv:2307.07635. https://arxiv.org/abs/2307.07635

Ke, L., Ye, M., Danelljan, M., Liu, Y., Tai, Y.-W., Tang, C.-K., & Yu, F. (2023). Segment anything in high quality. arXiv preprint. arXiv:2306.01567. https://arxiv.org/abs/2306.01567

Khan, S., Naseer, M., Hayat, M., Zamir, S. W., Khan, F. S., & Shah, M. (2021). Transformers in vision: A survey. *ACM Computing Surveys (CSUR)*, *54*(10s), 200. https://dl.acm.org/doi/abs/10.1145/3505244

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint. arXiv:1412.6980. https://arxiv.org/abs/1412.6980

Kirillov, A., Girshick, R., He, K., & Dollár, P. (2019). Panoptic feature pyramid networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 6399–6408). https://openaccess.thecvf.com/content_CVPR_2019/html/Kirillov_Panoptic_Feature_Pyramid_Networks_CVPR_2019_paper.html

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

Li, X., Wei, T., Chen, Y. P., Tai, Y.-W., & Tang, C.-K. (2020). FSS-1000: A 1000-class dataset for few-shot segmentation. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA (pp. 2869–2878).

Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2117–2125). https://openaccess.thecvf.com/content_cvpr_2017/html/Lin_Feature_Pyramid_Networks_CVPR_2017_paper.html

Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Li, C., Yang, J., Su, H., & Zhu, J. (2023). Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. arXiv preprint. arXiv:2303.05499. https://arxiv.org/abs/2303.05499

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. arXiv preprint. arXiv:1907.11692. https://arxiv.org/abs/1907.11692

Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel (pp. 807–814).

Özgenel, Ç. F. (2019). *Concrete crack segmentation dataset* (Version 1). [Data set]. Mendeley Data. https://data.mendeley.com/datasets/jwsn7tfbrp/1

Pan, X., Yang, T., Xiao, Y., Yao, H., & Adeli, H. (2023). Vision-based real-time structural vibration measurement through deep-learning-based detection and tracking methods. *Engineering Structures*, *281*, 115676.

Pan, X., & Yang, T. Y. (2023). 3D vision-based bolt loosening assessment using photogrammetry, deep neural networks, and 3D point-cloud processing. *Journal of Building Engineering*, *70*, 106326.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., & Lerer, A. (2017). *Automatic differentiation in PyTorch*. https://openreview.net/forum?id=BJJsrmfCZ

PyTorch, T. (2023). *Accelerating generative AI with PyTorch: Segment anything, fast*. https://pytorch.org/blog/accelerating-generative-ai/#:~:text=As%20announced%20during%20the%20PyTorch,Torch

Qarib, H., & Adeli, H. (2016). A comparative study of signal processing methods for structural health monitoring. *Journal of Vibroengineering*, *18*(4), 2186–2204.

Rafiei, M. H., & Adeli, H. (2018). A novel unsupervised deep learning model for global and local health condition assessment of structures. *Engineering Structures*, *156*, 598–607.

14678667, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/mice.13211 by University Of British Columbia, Wiley Online Library on [21/04/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Ren, T., Liu, S., Zeng, A., Lin, J., Li, K., Cao, H., Chen, J., Huang, X., Chen, Y., & Yan, F. (2024). Grounded SAM: Assembling open-world models for diverse visual tasks. arXiv preprint. arXiv:2401.14159. https://arxiv.org/abs/2401.14159

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. In N. Navab, J. Hornegger, W. Wells, & A. Frangi (Eds.), *International conference on medical image computing and computer-assisted intervention* (pp. 234–241). Springer. https://link.springer.com/chapter/10.1007/978-3-319-24574-4_28

Sajedi, S. O., & Liang, X. (2020). Vibration-based semantic damage segmentation for large-scale structural health monitoring. *Computer-Aided Civil and Infrastructure Engineering*, *35*(6), 579–596.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT (pp. 4510–4520).

Schonberger, J. L., & Frahm, J.-M. (2016). Structure-from-motion revisited. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV (pp. 4104–4113).

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-CAM: Visual explanations from deep networks via gradient-based localization. Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy (pp. 618–626).

Shi, J., Yan, Q., Xu, L., & Jia, J. (2015). Hierarchical image saliency detection on extended CSSD. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *38*(4), 717–729.

Shi, Y., Cui, L., Qi, Z., Meng, F., & Chen, Z. (2016). Automatic road crack detection using random structured forests. *IEEE Transactions on Intelligent Transportation Systems*, *17*(12), 3434–3445.

Shihavuddin, A., & Chen, X. (2018). *DTU-Drone inspection images of wind turbine*. https://orbit.dtu.dk/en/publications/dtu-drone-inspection-images-of-wind-turbine

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint. arXiv:1409.1556. https://arxiv.org/abs/1409.1556

Sun, C., Shrivastava, A., Singh, S., & Gupta, A. (2017). Revisiting unreasonable effectiveness of data in deep learning era. Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy (pp. 843–852).

Tan, M., & Le, Q. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. *International Conference on Machine Learning*, *PMLR*, Long Beach, CA (pp. 6105–6114).

Tomasi, C., & Kanade, T. (1991). Detection and tracking of point features. *International Journal of Computer Vision*, *9*, 137–154.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., & Jégou, H. (2021). Training data-efficient image transformers & distillation through attention. *International Conference on Machine Learning*, *PMLR*, Virtual Event (pp. 10347–10357).

Touvron, H., Vedaldi, A., Douze, M., & Jégou, H. (2019). Fixing the train-test resolution discrepancy. *Advances in Neural Information Processing Systems*, *32*, Vancouver, Canada.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems, *30*, Long Beach, CA.

Westoby, M. J., Brasington, J., Glasser, N. F., Hambrey, M. J., & Reynolds, J. M. (2012). 'Structure-from-motion' photogrammetry: A low-cost, effective tool for geoscience applications. *Geomorphology*, *179*, 300–314.

Yang, C., Zhang, L., Lu, H., Ruan, X., & Yang, M.-H. (2013). Saliency detection via graph-based manifold ranking. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR (pp. 3166–3173).

Ye, L., Rochan, M., Liu, Z., & Wang, Y. (2019). Cross-modal self-attention network for referring image segmentation. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA (pp. 10502–10511).

Zhang, C., Han, D., Qiao, Y., Kim, J. U., Bae, S.-H., Lee, S., & Hong, C. S. (2023). Faster segment anything: Towards lightweight SAM for mobile applications. arXiv preprint. arXiv:2306.14289. https://arxiv.org/abs/2306.14289

Zhang, H., Goodfellow, I., Metaxas, D., & Odena, A. (2019). Self-attention generative adversarial networks. *International Conference on Machine Learning*, *PMLR*, Long Beach, CA (pp. 7354–7363).

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

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2016). Learning deep features for discriminative localization. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV (pp. 2921–2929).

Zhou, Z., Rahman Siddiquee, M. M., Tajbakhsh, N., & Liang, J. (2018). UNet++: A nested U-Net architecture for medical image segmentation. *Deep learning in medical image analysis and multimodal learning for clinical decision support: 4th international workshop, DLMIA 2018, and 8th international workshop, ML-CDS 2018, held in conjunction with MICCAI 2018* (pp. 3–11). Springer. link.springer.com/chapter/10.1007/978-3-030-00889-5_1

How to cite this article: Azimi, M., & Yang, T. Y. (2024). Transformer-based framework for accurate segmentation of high-resolution images in structural health monitoring. *Computer-Aided Civil and Infrastructure Engineering*, 1–15. https://doi.org/10.1111/mice.13211

14678667, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/mice.13211 by University Of British Columbia, Wiley Online Library on [21/04/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License