Deep Neural Network based Visual Inspection with 3D Metric Measurement of Concrete Defects using Wall-climbing Robot

Liang Yang[1,2], Bing Li [1], Guoyong Yang [2], Yong Chang [2], Zhaoming Liu [2], Biao Jiang [3], Jizhong Xiao[1,*]

***Abstract*—This paper presents a novel inspection method using a deep neural network called InspectionNet to detects the crack and spalling defects on concrete structures performed by a novel wall-climbing robot. First, we create a pixel-level semantic dataset which includes 820 labeled images. The training on the dataset for InspectionNet is performed with $12,000$ iterations for each defect type. Second, we propose an inspection method to obtain 3D metric measurement by using an RGB-D camera-based visual simultaneous localization and mapping system (SLAM), which is able to generate pose coupled key-frames with depth information. Therefore, the semantic inspection results can be registered in the concrete structure 3D model, which provides metric information for condition assessment and monitoring. Third, we present our new generation wall-climbing robot to perform the inspection task on both horizontal and vertical surfaces. Our field experiments demonstrate that our wall-climbing robot and inspection system can perform robust 3D metric inspection even in a low illuminated environment.**

***Index Terms*—Wall-climbing robot, visual inspection, deep learning, concrete structure spalling and crack dataset, semantic 3D reconstruction**

I. INTRODUCTION

STRUCTURAL health monitoring (SHM) plays a significant role on performance evaluation and condition assessments for the Nation's highway transportation assets, and it can promote its operational safety and longevity based on data-driven analysis and decisions. The Federal Highway Administration (FHWA) of the U.S. Department of Transportation (DOT) has launched the Long-Term Bridge Performance (LTBP) Program in 2015 to facilitate the SHM by collecting critical performance data [1]. According to the FHWA's latest bridge element inspection manual [2], New York Bridge Inspection Manual [3], and Tunnel Operations, Maintenance, Inspection, and Evaluation (TOMIE) Manual [4], it is required to identify, measure, and record the condition state information during a routine inspection on bridges and tunnels. Such condition states include spall (delamination, patched area), exposed rebar, cracking, abrasion (wear), and damage, etc. In this research, we introduce a data-driven visual inspection robot for spalling (with exposed rebar or not) and cracking inspection. The spalling and cracks are the main factors affecting the condition states of reinforcing concrete [5].

Automated Visual inspection [6], [7], [8] has become a popular approach for structural surface inspection with the advance development in optics devices, such as for structural displacement measurement, crack or spalling inspection, and strain or stress monitoring. Several robotics inspection systems have been developed for automated visual inspection data collection and processing. Researchers in Rutgers University developed a mobile robotic crack inspection and mapping system, and it uses edge detection algorithm to detect the cracks on concrete bridge decks and generate the crack map for bridge maintenance [9]. Under the support of FHWA LTBP program, an autonomous bridge deck inspection mobile robotic system has been developed with visual cameras and other detection sensors [10], [11]. Unmanned aerial vehicle (UAV) has also been deployed for bridge visual inspection [12]. Our wall-climbing robots provide vertical mobility to perform visual inspection and GPR-based subsurface flaw inspection on both vertical and horizontal surfaces [13].

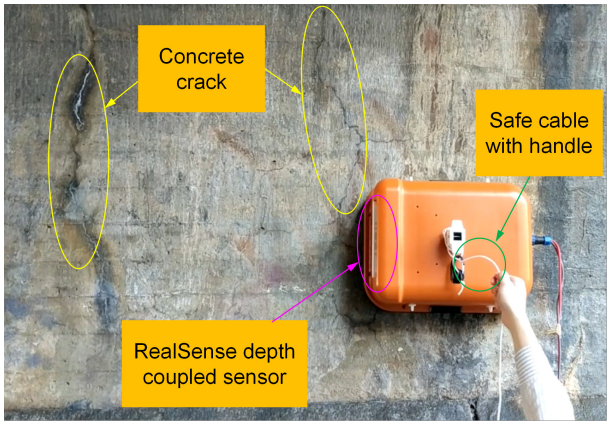


Fig. 1. Proposed wall-climbing inspection robot on the vertical surface of a bridge-tunnel at Riverside Dr W 155th St, New York, NY 10032.

Various image processing algorithms have been explored for concrete structures surface crack and spalling inspection. As an earlier work, entire image processing procedures were presented by Oh [14], and it used a median filter, morphological operations and intensity gradient for crack detection. A gray-scale histogram analysis and automatic peaks detection approaches were also used for concrete surface images inspection [15]. A crack-defragmentation approach of fragment grouping and fragment connection was presented by Wu, and an artificial neural network (ANN) was used for crack detection classification [16]. More recently, convolutional neural network (CNN) has been deployed for crack classification

This work was supported by University Transportation Center on INSpecting and Preserving Infrastructure through Robotic Exploration (INSPIRE), with USA Federal High Way Administration (FHWA) grant FAIN 69A3551747126. *Corresponding author.

[1]The CCNY Robotics Lab, Electrical Engineering Department, The City College of New York, New York, USA lyang1,bli,jxiao@ccny.cuny.edus

[2]University of Chinese Academy of Sciences, Shenyang Institute of Automation, Chinese Academy of Sciences gyyang,changyong, liuzhaoming@sia.cn

[3] Department of Natural Sciences, Hostos Community College, New York, NY, USA bjiang@ccny.cuny.edu

on concrete structure images [17]. However, towards data-driven visual inspection of concrete structures, there are still some challenges needed to be solved. 1) high-quality dataset with labeling for detection model training and ground truth verification; 2) semantic segmentation with depth augmentation; 3) accurate positioning, registration and visualization of the detected flaws, 4) 3D reconstruction and modeling of inspection results; 5) the ability to perform automated inspection on both horizontal surfaces (i.e., bridge deck) and vertical surfaces (e.g., bridge foundation).

In this paper, we introduce our new generation of the wall-climbing robot system, as shown in Fig.1. The system integrates multiple hardware modules within a compact robot body, including motion control, negative pressure module, RGB-D camera with pan-tilt mechanism for visual inspection, and ground penetration radar (GPR) for delamination detection, as well as visual odometry positioning and 3D mapping software. This system aims at providing a complete automated visual inspection data collection and analysis approach with semantic segmentation based on 3D reconstruction mapping to solve the above-mentioned challenges.

In addition, we create a high-quality concrete structure spalling and crack (CSSC) dataset for deep learning visual inspection and propose an InspectionNet connected neural network for semantic segmentation. Compared with our previous work [17], we further augmente our dataset with pixel-level labeling. We released this unique dataset publicly [1] and open source code in GitHub [2] for the visual inspection research communities. As the pioneer dataset for concrete visual inspection, CSSC dataset includes pixel-level labeled 522 crack images and 298 spalling images and over $10,000$ field images.

II. INSPECTION WALL-CLIMBING ROBOT DESIGN

In this section, we briefly introduce our new generation wall-climbing robot platform, including the mechanical system and adhesion module design, control system design, and the inspection sensors installed inside the robot body.

A. Wall-climbing Robot Mechanism

The mechanical model of the wall-climbing robot is shown in Fig. 2 and all the components are mounted on the main chassis. The drive train consists of two drive wheels in the rear and one omni-direction wheel in the front, and those wheels support the chassis for locomotion. The drive wheel is covered with soft rubber tread to increase the friction force between the wheel and the wall, and two brushless DC motors are deployed for the mobility. A bumper is connected to the chassis with two springs to hold its position. A micro-switch is triggered when the robot bumper hits an obstacle or reaches a corner. A RealSense RGB-D Camera is mounted on the robot, and it can be driven by a servo to change view angle. A pressure sensor is also mounted to monitor the pressure level inside the vacuum chamber. All the other components such as the power over Ethernet (POE) module, motor driver, wheel motor, DC-DC converter, digital signal processing (DSP) control board and Intel NUC board are installed on the top of the chassis.

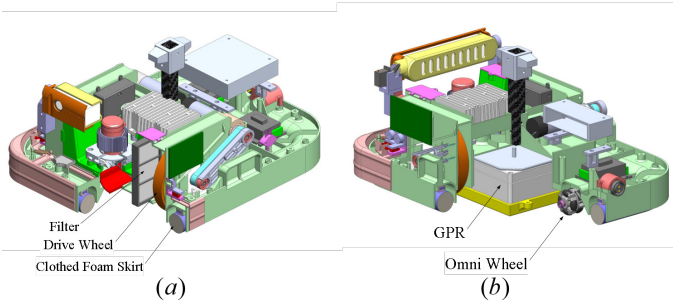


Fig. 2. Wall-climbing system complete design. (a) We re-design the skirt for the vacuum to ensure a higher detaching force. (b) The GPR is mounted in the body of the wall-climbing robot, and the driving part of the robot is consists of differential drive and omni-wheel.

Negative pressure adhesive method (NPAM) is applied for the wall-climbing robot to be attached to the vertical surface. The vacuum chamber is enclosed by the chassis and the clothed foam skirt which is a foam tube coated with parachute cloth to reduce the friction force and increase the persistence. Parachute cloth also connects the foam tube to a thin frame which is fixed underneath the chassis. So the clothed foam skirt is more flexible to seal the chamber when climbing on the uneven wall. The NPAM is installed onto the upper surface in the front section of the chassis, where there is a hole go through to the chamber under the chassis. Two filters are used to protect the NPAM from the debris sucked into the vacuum motor.

B. Interactive User Control Interface

For the remote control and visual monitoring purpose, we design an wireless control software (Android platform) for the wall-climbing robot. An illustration of the wall-climbing robot remote control and monitoring GUI is illustrated in Fig.7. The remote controller provides the control functions for the vacuum suction motor with a linear adjustable scroller, wheel motors and mobility speed control, LED light compensation control, and camera streaming video for monitoring.

C. Robotics Visual SLAM

The goal of the wall-climbing robot inspection system aims at providing quantitative measurement of defects, including high precision information on area, width and depth of cracks, and the location of the defects in reference to the starting position. We develop a visual SLAM system which enable the wall-climbing robot to continuously update the position of the robot (via the on-board camera) with respect to a world coordination system (Xw,Yw,Zw), which is defined as East-North-Up (x axis points East, y points North and z up).

In this paper, the visual SLAM is performed through a pose-graph approach which is proposed based on our previous visual-odometry [18], with an RGB-D sensor to assist fast and accurate depth acquiring. Let denote a relative transformation

[1]https://robotics.ccny.cuny.edu
[2]https://github.com/ccny-ros-pkg

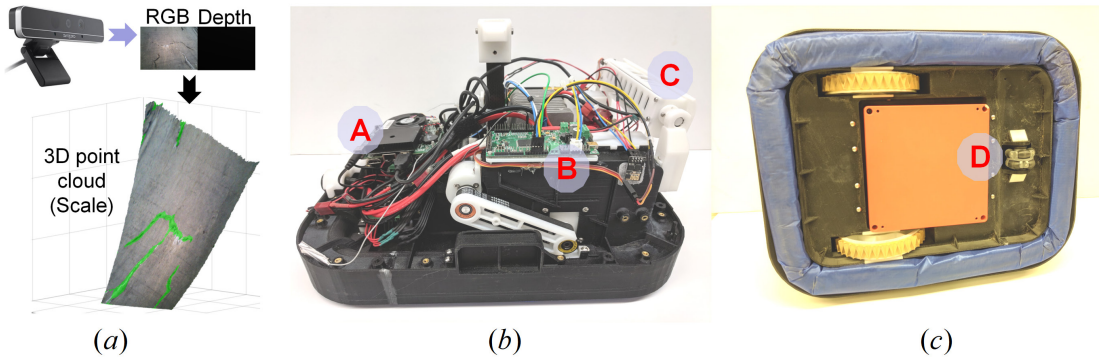


Fig. 3. The inspection wall-climbing robot and sensor suite. (a) An RGB-D camera is used to acquire image and scale information for inspection. (b) The robot whole control system layout. A: The Intel NUC on-board computer to process sensory information. B: The on-board control system, which is a DSP-based controller board. C: Realsense RGB-D camera for visual inspection and positioning. (c) The bottom side of the wall-climbing robot with a GPR sensor and differential drive wheels. D: The GPR sensor with 2.4 GHz frequency.

between two consecutive frames as $R \in SO(3)$ as (where $SO(3)$ special Lie rotation group), and $t \in R^3$ denotes the translation in the predefined world coordinate system. Then, the step-motion-estimation of two frames I_p and I_q with corresponding features F_{I_p}, F_{I_q} can be represented as:

$$\{R,t\} = \arg\min_{R,t} \sum_{i\in\{1,\ldots,N\}} L_\rho(F_{I_p}(i) - \pi(||R \cdot F_{I_q}(i) + t)||^2_\Sigma) \quad (1)$$

where arg min denotes the linear regression process toward minimal, $L_\rho(\cdot)$ is the Huber loss based cost function, and $||_\Sigma$ denotes the covariance weighted sum toward a robust convergence. Then, we can obtain the pose P^I for each frame through a cumulative approach.

Like most SLAM system [19], [20] of performing sparse generic linear optimization for pose [21] estimation, we also deploy this approach via performing local and global optimization to decrease drift. For SLAM, the environment reconstruction can be done via using key-frames $\{I^K, P^k\}$. Once a new frame is obtained, two processes have to be executed: 1) execute loop closure detection based on feature matching through Bag of Words (locally and globally); 2) key-frame evaluation. For two correlated frames $I^i and I^j$ which has common features F_{ij} as a weight for pose-graph, the goal is to optimize the projection with the transformation:

$$\{F, I^L, T\} = \arg\min_{X^i\in F, \{R_i,t_i\}\in T} L_\rho ||^r x(\cdot) - \pi(R^c_i x + t_i)||^2_\Sigma \quad (2)$$

where I^L denotes all the local frames that can see features F, T denotes the transformation between frames, $^r x(\cdot)$ denotes the 2D features of the reference frame if given current frame $^c x$.

D. Inspection Sensors

There are two inspection sensors equipped with our wall-climbing robot. 1) RGB-D camera for depth-aided visual inspection: performs metric (size and position) inspection suing deep learning and SLAM. 2) GPR sensor for subsurface flaw detection. In this paper, we focus on the visual inspection using the RGB-D sensor and cutting-edge image processing approach.

III. DEEP VISUAL INSPECTION

In order to perform pixel-level segmentation for concrete defects, we first propose a fully convolutional neural network called InspectionNet for crack and spalling semantic segmentation. Then, the detection results are registered in the 3D model to obtain metric information such as size and area. To the best of our knowledge, this is the first time for visual inspection registered on the 3D model for complete semantic visualization and SHM diagnosis.

A. Visual Inspection: Dataset and Training

Basic Knowledge and Proposed Method: The detection approach introduced in this paper is proposed based on FCN-8s[22] and U-net [23], and these two deep-neural-networks perform end-to-end full pixel level segmentation. Current researches on semantic segmentation all share the same merits of deploying a multi-level fusing structure to fuse multiple scale response. Thus, these semantic segmentation networks can use the mid-level Gestalt law information. The proposed architecture for spalling/crack segmentation is represented in Fig.4, where we adopt the U-net style structure to form the skeleton of the network. Based on our experience, the network has to obtain lower layers' information to perform a higher order of edge segmentation. Unlike U-net, we directly deploy the VGG-16 as the pre-feature-extraction layers and also keep the resolution as well. Also, the left deconvolutional layers perform concatenation with the left corresponding convolutional layers, and output after one convolutional operation.

For image segmentation, given N pairs of input training data $Tr = \{(X_m, Y_m), m = 1,\ldots,M\}$, where X_m denotes the original input image with three-dimensional array of size $w \times h \times d$, where h and w are width and height, and d is the channel dimension. $Y_n = \{y^m_i, i = 1,\ldots,|X_m|\}$, $y^m_i \in N$ (N is natural numbers) denotes the corresponding ground truth segmentation map for image X_m with a total number of $|X_m|$ pixels with one channel. The purpose of the training is to find a network which can learn to form a mapping from input to ground truth.

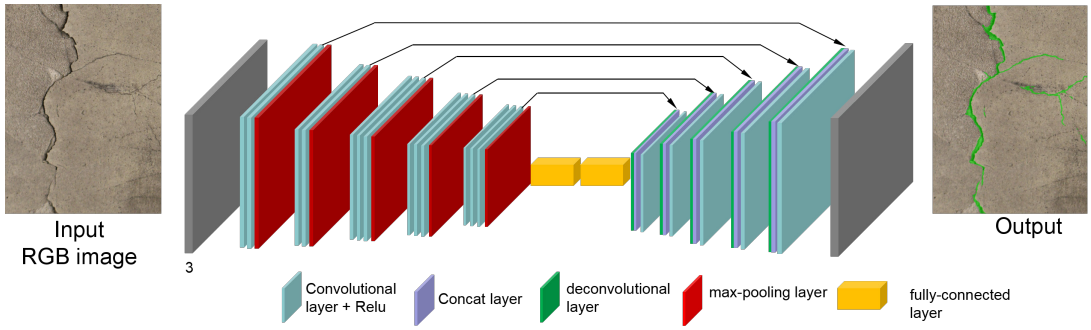


Fig. 4. An illustrated of the proposed crack and spalling detection network. Inspection net consists of a total 26 layers, and we introduce deconvolutional layers to do upsampling on the right side. The first 16 layers are based on VGG-16, and we perform transfer learning by initializing the network using VGG-16 weight.

(*a*) *original image* (*b*) CSSC Label (*c*) *Our semantic annotation*

Fig. 5. (a) concrete spalling is the failure that concrete breaks down into small spalls from the concrete body. (2) The label from CSSC dataset. (3) Our newly labeled image using a closed polygon to describe the spalling part.

Dataset and Labeling: The dataset to be annotated is provided by Liang Yang et al [17], called Concrete Structure Spalling and Crack (CSSC) database. However, the spalling image in CSSC was initially proposed to do region-based classification based on fine-tuned VGGNET [24]. This paper performs further annotation on the dataset to do semantic segmentation. We define the following guidelines to be keys to perform high-quality annotation: (1) only concrete spalling and crack meaningful regions should be annotated; (2) annotation only perform at target spalling and crack region, other regions should be annotated as background. (3) the spalling region should be annotated with independent polygons; (4) crack region should be annotated in pixel level detail, especially for unclear crack. These guidelines enable us to label crack and spalling carefully with great details.

(1) *Spalling annotation*: Previous CSSC dataset is only labeled with exposed rebar (as illustrated in Fig.5.b). In this paper, we use Labelme [25] to do the spalling labeling. We name the spalling region as 'spalling', and each annotator is asked to follow the definition provided by civil engineers to label the corresponding spalling region. The annotation only performs on such region which can be named as spalling, where the boundaries should be able to provide a clear comparison. Thus, multiple polygons can exist for spalling in one image, and we name the other regions as background (Fig.5.c). Finally, we further process the labels to generate expected ground truth data.

(2) *Crack annotation* Crack region tends to more scale variant and with low contrast, and the CSSC dataset which already provides 104 labeled images. Annotators are asked to label the minor crack regions over all the images with a semantic name tag. Of particular importance is that if a crack region is blurred, the visible crack regions should all be annotated.

It should be noted that the dataset annotation is done by using Labelme [25] and Adobe Photoshop, where Labelme for spalling area labeling and Adobe Photoshop for crack labeling.

Training and Analysis

The data for training consists of the original color image and the pixel level labeled image. Same as [23], we deploy one-hot encoding for image segmentation classification for our spalling and crack. To achieve robust model, we perform image flipping, rotation, and sub-cropping to do image augmentation for training as well as validation.

This paper defines the loss as the pixel-wise soft-max which is introduced in U-net [26] to perform loss calculation over predicted feature map with given ground truth. For the given image set $X = \{X_m | X_m = \{x_i^m, i = 1, ..., |X_m|\}, m = 1, ..., M\}$, the soft-max is defined as

$$p_k(x_i^m) = exp(a_k(x_i^m))/(\sum_{k'=1}^{K} exp(a_{k'}(x_i^m))) \quad (3)$$

where $a_k(x_i^m)$ denotes the activation at feature channel k at pixel position x_i^m, K denotes the number of clusters, $p_k(x_i^m)$ denotes the maximum function. Then, the loss based on the cross entropy is defined as

$$E = \sum_{x_i^m \in X_m} w^U(x) log(p_{(k)(x_i^m)}(x_i^m)) \quad (4)$$

where $(k) \in \{1, ..., K\}$ denotes the true label of each pixel, and w^U is the importance weight of each pixel.

The cross evaluation of the model is performed every 200 iterations, we compare with the following perspectives: 1) F1 score: $F1 = 2 * (precision * recall)/(precision + recall)$; 2) average precision to indicate the average pixel-wise accuracy of the evaluation: $AP = Truepositive/(Truepositives + Falsepositive)$. We also enable the visualization of the cross-entropy loss as well the training precision.

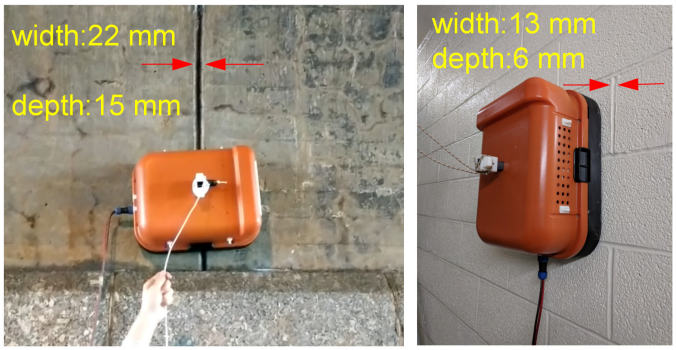
B. 3D Metric Registration

Civil engineers [3] perform the bridge or tunnel inspection by analyzing the perspectives: 1) location: the locations of

TABLE I
ADHESIVE FORCE TESTING ON GROUND AND VERTICAL SURFACES

Adhesive motor speed (percentage %)	15	20	25	30	35	40	45	50
Ground pull-up adhesive force (Kg)	4.62	5.62	8.22	11.02	16.42	24.5	28.95	31.62
Vertical pull-down load adhesive force (Kg)	N/A	1.3	4.5	6.4	8.0	9.7	10.4	10.5

cracks and spallings in reference to the bridge or tunnel structure elements. 2) type: the type of defects, such as crack, spalling, erosion, etc. 3) size: the metric information which related to width, depth, and area.4) quantity: the number of defects. 5) severity of the deterioration and deficiencies: the overall condition state which has to be evaluated by a qualified inspector according to the size, type and location information of the defects. It means that the image-based pixel evaluation is not enough, whereas the quantitative metric measurement is needed to perform a scalable evaluation of structure health.



(a)Under - bridge area test (b)Test at CCNY

Fig. 6. Proposed wall-climbing robot with ditches crossing testing

The visual SLAM system as we discussed in Section II, performs positioning for each inspection frame (i.e., images data associated with SLAM positioning data of the wall-climbing robot). The inspection frame has the format as $\{I, P\}$, where I denotes the image and P denotes the corresponding pose (position and orientation). The InspectionNet allows us to prediction ${}^{c}X_m = x_i | x_i \in class(c)$, where ${}^{c}X_m$ denotes the pixels which belong to class c (can be spalling or crack). The information of ${}^{c}X_m$ is pixel level position information. Then, the corresponding depth information can be used to register the defect area into the 3D space with camera intrinsic parameters. Finally, the 3D defect area information is used to do the metric evaluation, such as area calculation, crack width evaluation.

IV. EXPERIMENT

The mobility and payload tests of our wall-climbing robot were performed in the Steinman Hall (ST) at The City College of New York (CCNY). The robot was commanded to maneuver in both ground and vertical surfaces. To evaluate our deep visual inspection, a high-quality CSSC dataset was further built for segmentation detection model training. Field tests and evaluation of the robotic system with deep visual inspection were conducted on the vertical wall of a bridge tunnel at Riverside Dr W 155th St, New York, NY 10032, as shown in Fig. 1.

A field test demo of InspectionNet result and our inspection robot is as shown in demo video [3].

A. Robot Mobility and Adhesion Test

Our wall-climbing robot is a compact device with dimension of 16.5 inches $\times$ 13 inches $\times$ 8 inches, and with self-weight 12 lbs. The maneuverability tests of the robot were conducted in various type of vertical walls, from smooth surfaces to rough concrete surfaces. The vacuum chamber is sealed by clothed foam skirt to provide persistent adhesion based on negative pressure. Therefore, our wall-climbing robot can cross over small ditches while moving on vertical walls. Fig. 6 (a) shows our robot crosses a ditch with width 22mm and depth 15mm during the visual inspection of the bridge-tunnel surfaces. Fig. 6 (b) shows our robot crosses a ditch with width 13mm and depth 6mm on the wall of Steinman Hall in CCNY.

The inspection robot is controlled by a remote controller on Android platform, as shown in Fig. 7. On the left, the *vacuum speed control tab* provides the speed control for the vacuum suction motor, and the value is adjustable in the percentage of the maximum speed. The *LED control tab* controls the LED lighting for visual inspection in dark area, which is especially important for most tunnel inspections. The *motion control tab* perform motion control, the four direction tabs allow forward, backward, right turn, and left turn motion. Its speed can be adjusted by the *speed pop-up scroller*. The middle *image monitoring window* shows the real-time image streaming from the on-board camera. On the right, the *camera view angle control tab* allows us to change the view angle of the RGB-D camera, thus provides the best view of the defect area.

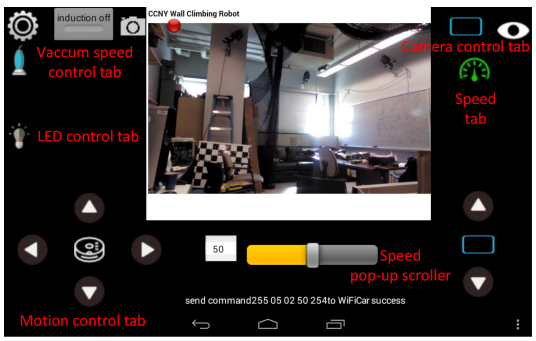


Fig. 7. The Screen Shot of the remote GUI interface

We conduct the adhesive force testing on the ground and smooth vertical surfaces, and the results are shown in Table

[3]https://tinyurl.com/3DInspectionRobot

TABLE II
CROSS VALIDATION SPEED DURING TRAINING(FRAMES/SECOND)

Item %	Crack Tr_1	Crack Tr_2	Crack Tr_3	Spall Tr_1	Spall Tr_2
Processing speed (frames per second)	6.5000	7.9997	6.2524	6.5000	6.4698

I. By selecting various adhesive suction motor speed (from 15% to 50%) in the unit of percentage with regarding to the maximum speed, we measured the pull-up adhesive force when the robot is placed on the ground surface, and the force is ranging from 4.62 to 31.62 Kg. In addition, we measured the vertical pull-down adhesive force (carrying load), and the force is ranging from 1.3 to 10.5 Kg. For the adhesive suction motor speed lesser than the minimum attaching speed, its carrying load is marked as not available.

B. Model Training Analysis

Dataset Based on the CSSC dataset, in which 278 spalling images with exposed rebar labeling and 954 crack image with 104 labeled images, we further expand spalling images to 298. For training purpose, we have a total 298 spalling image with pixel level labeling, and 522 crack images with pixel level labeling. In addition to the original labeled images, we further cropped the large image size to a maximum of $1,600 \times 1,100$, and we also perform flipping to augment the images. Then, we get a total of $4,473$ images for the crack model, where $3,147$ images for training, 498 for cross-validation, and 828 for testing. The spalling detection model has 627 for training, 90 for cross-validation, and 177 for testing.

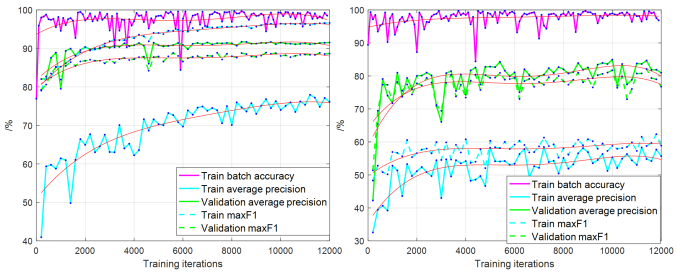
Inspection performance of both crack and spalling model are measured using batch concurrent accuracy, average precision, and max F1 score [27].

Crack Training

Since the InspectionNet is built on VGG-16, the right side parameters of de-convolutional layers and convolutional layers are randomly initialized. Thus the transfer learning ensures a fast convergence. The training has a total $12,000$ iteration. For crack inspection, we found that FCN-8s is not able to detect the crack. However, we compare the performance of the inspect net between training using partial and complete dataset we have. For the partial training dataset, we use the CSSC dataset original 104 images to build the training and testing dataset. It is illustrated in Fig.8 that the raw concurrent batch accuracy can reaches 95% within $1,000$ iterations. However, as one can see in the graph that the InspectionNet can only reach 81.4% average precision of complete dataset compared to 91.5% of performing training on whole 522 images generated dataset. For loos and entropy, as shown in Fig.9, the partial training dataset can lead to faster convergence. In this graph, it also shows that the loss of performing complete dataset using inspection-net is harder to converge.

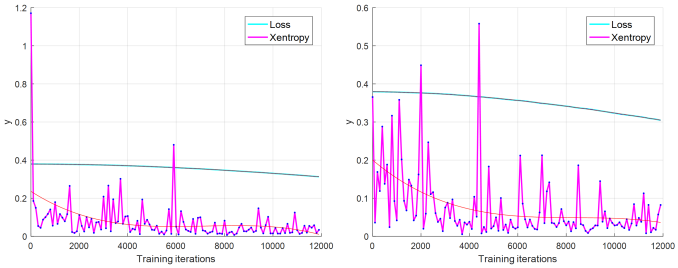
Spalling Training

Spalling training is also executed in $12,000$ steps, and also in a transfer learning approach of using the VGG-16 model parameter to initialize the InspectionNet. To provide a baseline for the spalling detection, we compared the performance of



(a)Training with partial dataset (b)Training with complete dataset

Fig. 8. A comparative illustration of performs crack inspection training using (a) partial dataset with 104 images. (b) the complete dataset with 522 labeled images, and Batch concurrent accuracy, average accuracy, and max F1 score are compared for the two cases.



(a)Training with partial dataset (b)Training with complete dataset

Fig. 9. The comparison of training loss and entropy for (a) partial dataset and (b) the complete dataset.

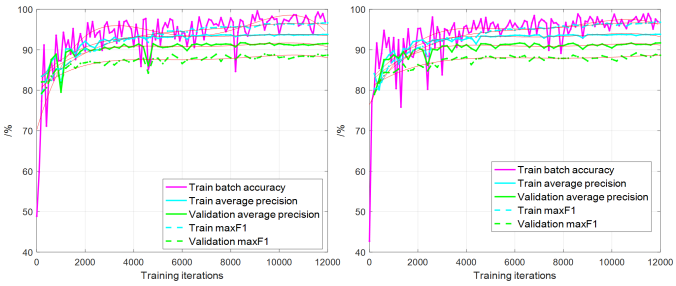
InspectionNet with the FCN-8s net. The comparative result of average precision, the max F1 score is represented in Fig.10. As one can see that both models can achieve an average precision over 90% within $4,000$ iterations, and the InspectionNet for spalling detection is only 1% higher compared with FCN-8s. For the long-term performance, we can see that the InspectionNet is more stable compared with FCN-8s since our InspectionNet has a higher order feature information to assist residue passing. The loss and entropy of InspectionNet and FCN-8s is illustrated in Fig.11, where InspectionNet converges faster than the FCN-8s model.

Processing Speed

To evaluate the processing speed, we perform 5 times of testing. We perform 3 times of crack detection evaluation and 2 times of spalling detection evaluation, and we calculate the mean processing speed of each session. It is illustrated in Table.II that our InspectionNet has a min average 6.2 frames per second.

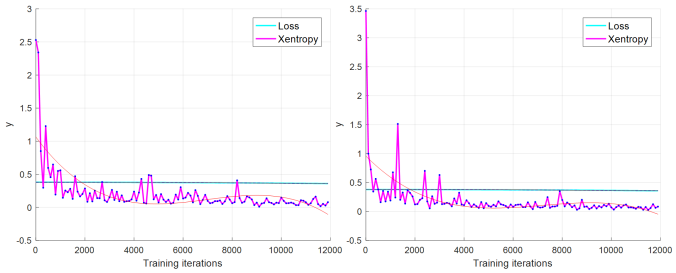
C. Dataset Test and Field Test

The evaluation of the visual inspection system is performed in two steps. First, we test the detection performance on



(a)Fcn - 8s model (b)Inspection net training

Fig. 10. Comparison of training between (a) FCN-8s model and (b) our inspection model. We compare the perspectives including batch concurrent accuracy, average accuracy and max F1 score.



(a)Fcn - 8s model (b)Inspection net training

Fig. 11. The comparison of training loss and entropy for (a) FCN-8s model and (b) our InspectionNet.

the test dataset and evaluate the average accuracy. In the second step, we perform field tests under a bridge with semantic reconstruction. In the field tests, we consider both normal illumination and low illumination situation to perform inspection and 3D reconstruction.

The performance of performing detection on CSSC dataset is illustrated in Fig.12. In this figure, D_T1 denotes test on the dataset, where $D_T1:(1)\ (5)$ are spalling detection result and $D_T1:(5)\ (10)$ are crack detection result. For crack detection on the dataset, we have an average precision of 76.41%. The average precision of spalling detection is 87.9319%. F_T1 and F_T2 denotes to set of tests. $F_T1:(1)\ (10)$ illustrate that the InspectionNet can perform detection very well on field data, where the minor cracks can be easily segmented out. $F_T2:(1),(3),(6),and(8)$ indicate the segmentation of original image for defects. $F_T2:(2),(4),(7),and(9)$ are the corresponding model original output. $F_T2:(5)and(10)$ are the heat-map point cloud maps for defect regions.

3D Metric Semantic Registration

Our goal is to perform metric semantic reconstruction, and we perform two tests which are represented in Fig.13 and Fig.14. The 3D reconstruction is performed by coupling the image frames with *pose* and *time*, where the frames are key-frames for SLAM. Then, the InspectionNet detects the region of defects as described in Section III.B. Thus we can register to 3D space with the semantic labeled image. It is illustrated in Fig.13, the green area (Fig.13.(a)) and blue area (Fig.13.(b)) denote the defect regions, where all the information are in 3D with true scale. Then the civil engineer can retrieve the detects region to calculate the defect region area and width or depth information. We further test the system in low illumination area as illustrated in Fig.14. The representative 2D image detection results are $F_T2:(6)\ (10)$, and we can see that the SLAM and InspectionNet are both able to perform positioning and inspection under such situation.

V. CONCLUSION

This paper introduce our new generation wall-climbing robot system for concrete structure visual inspection. A state-of-the-art CSSC dataset with pixel-level labeling and an InspectionNet network were designed for semantic segmentation. Furthermore, based on our design on the visual odometry positioning and 3D reconstruction, the detected results were registered in the 3D model to provide metric information for concrete structure condition assessment. The field experiments show the effectiveness of our robot system for vertical mobility and visual inspection.

Future work will consider fully autonomous inspection on vertical surfaces, and integration of the visual inspection with subsurface flaw detection using GPR. The contact-based subsurface inspection is not able to performed by unmanned aerial vehicle (UAV). It is our interest to build a comprehensive inspection robot system solution for automated structural health assessment.

REFERENCES

[1] N. Gucunski, "Condition assessment of bridge deck using various nondestructive evaluation (nde) technologies," *LTBP*, vol. 5, pp. 1–7, 2015.
[2] F. H. Administration, "Specification for the national bridge inventory bridge elements," 2014.
[3] N. Y. D. of Transportation, "Bridge inspection manual," January, 2016.
[4] F. H. Administration, "Tunnel operations, maintenance, inspection, and evaluation (tomie) manual." 2015.
[5] C. Koch, K. Georgieva, V. Kasireddy, B. Akinci, and P. Fieguth, "A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure," *Advanced Engineering Informatics*, vol. 29, no. 2, pp. 196–210, 2015.
[6] G. Li, S. He, Y. Ju, and K. Du, "Long-distance precision inspection method for bridge cracks with image processing," *Automation in Construction*, vol. 41, pp. 83–95, 2014.
[7] R. Adhikari, O. Moselhi, and A. Bagchi, "Image-based retrieval of concrete crack properties for bridge inspection," *Automation in construction*, vol. 39, pp. 180–194, 2014.
[8] M. R. Jahanshahi and S. F. Masri, "Adaptive vision-based crack detection using 3d scene reconstruction for condition assessment of structures," *Automation in Construction*, vol. 22, pp. 567–576, 2012.
[9] R. S. Lim, H. M. La, and W. Sheng, "A robotic crack inspection and mapping system for bridge deck maintenance," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 2, pp. 367–378, 2014.
[10] P. Prasanna, K. J. Dana, N. Gucunski, B. B. Basily, H. M. La, R. S. Lim, and H. Parvardeh, "Automated crack detection on concrete bridges," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 591–599, 2016.
[11] H. M. La, N. Gucunski, K. Dana, and S.-H. Kee, "Development of an autonomous bridge deck inspection robotic system," *Journal of Field Robotics*, vol. 34, no. 8, pp. 1489–1504, 2017.
[12] N. Hallermann and G. Morgenthal, "Visual inspection strategies for large bridges using unmanned aerial vehicles (uav)," in *Proc. of 7th IABMAS, International Conference on Bridge Maintenance, Safety and Management*, 2014, pp. 661–667.
[13] B. Li, K. Ushiroda, L. Yang, Q. Song, and J. Xiao, "Wall-climbing robot for non-destructive evaluation using impact-echo and metric learning svm," *International Journal of Intelligent Robotics and Applications*, vol. 1, no. 3, pp. 255–270, 2017.

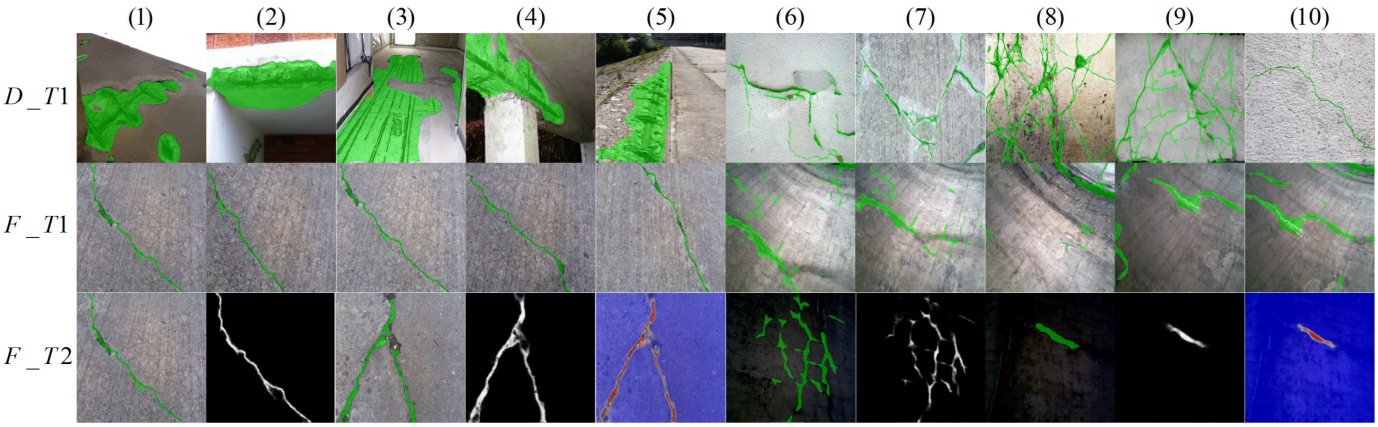
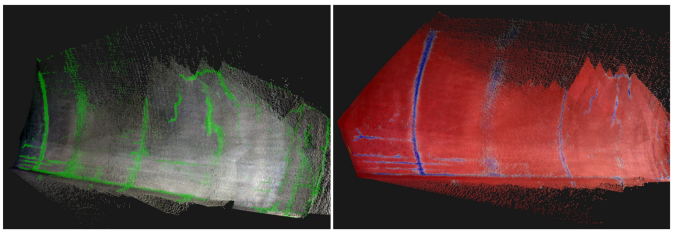


Fig. 12. An illustration of InspectionNet detection results on CSSC dataset and field collected data. The green color denotes the defects region of the original image, the red color is used to highlight the defect region, and the white and black image is the original output of the InspectionNet.



(a)Semantic defects labled point cloud map (b) Heat map like point cloud map

Fig. 13. The 3D reconstructed result of obtaining the 3D metric information. (a) is the 3D defects segmented point cloud map, (b) is the corresponding heat-map point cloud map, where blue denote the defect area.

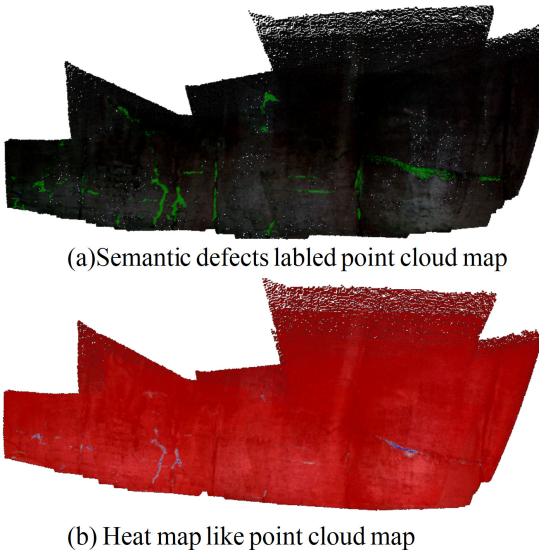


Fig. 14. The detection result registered in the 3D model, and it shows the robustness of our proposed approach even for low-illumination images

[14] J.-K. Oh, G. Jang, S. Oh, J. H. Lee, B.-J. Yi, Y. S. Moon, J. S. Lee, and Y. Choi, "Bridge inspection robot system with machine vision," *Automation in Construction*, vol. 18, no. 7, pp. 929–941, 2009.

[15] T. H. Dinh, Q. Ha, and H. La, "Computer vision-based method for concrete crack detection," in *Control, Automation, Robotics and Vision (ICARCV), 2016 14th International Conference on*. IEEE, 2016, pp. 1–6.

[16] L. Wu, S. Mokhtari, A. Nazef, B. Nam, and H.-B. Yun, "Improvement of crack-detection accuracy using a novel crack defragmentation technique in image-based road assessment," *Journal of Computing in Civil Engineering*, vol. 30, no. 1, p. 04014118, 2014.

[17] Y. Liang, L. Bing, L. Wei, L. Zhaoming, Y. Guoyong, and X. Jizhong, "Deep concrete inspection using unmanned aerial vehicle towards cssc database," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2017.

[18] I. Dryanovski, R. G. Valenti, and J. Xiao, "Fast visual odometry and mapping from rgb-d data," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 2305–2310.

[19] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[20] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *European Conference on Computer Vision*. Springer, 2014, pp. 834–849.

[21] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g 2 o: A general framework for graph optimization," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 3607–3613.

[22] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[23] M. Teichmann, M. Weber, M. Zoellner, R. Cipolla, and R. Urtasun, "Multinet: Real-time joint semantic reasoning for autonomous driving," *arXiv preprint arXiv:1612.07695*, 2016.

[24] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[25] "Labelme," *https://github.com/mpitid/pylabelme*, 2011.

[26] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2015, pp. 234–241.

[27] M. Everingham and J. Winn, "The pascal visual object classes challenge 2012 (voc2012) development kit," *Pattern Analysis, Statistical Modelling and Computational Learning, Tech. Rep*, 2011.