2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Macau, China, November 4-8, 2019

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

Liang Yang[1,3], Bing Li [2], Guoyong Yang [3], Yong Chang [3], Zhaoming Liu [3], Biao Jiang [4], Jizhong Xiao[1,*]

***Abstract*— This paper presents a novel metric inspection robot system using a deep neural network to detect and measure surface flaws (i.e., crack and spalling) on concrete structures performed by a wall-climbing robot. The system consists of four modules: robotics data collection module to obtain RGB-D images and IMU measurement, visual-inertial SLAM module to generate pose coupled key-frames with depth information, InspectionNet module to classify each pixel into three classes (back-ground, crack and spalling), and 3D registration and map fusion module to register the flaw patch into registered 3D model overlaid and highlighted with detected flaws for spatial-contextual visualization. The system enables the metric model of each surface flaw patch with pixel-level accuracy and determines its location in 3D space that is significant for structural health assessment and monitoring. The InspectionNet achieves an average accuracy of 87.64% for crack and spalling inspection. We also demonstrate our InspectionNet is robust to view angle, scale and illumination variation. Finally, we design a metric voxel volume map to highlight the flaw in 3D model and provide location and metric information.**

I. INTRODUCTION

Structural health monitoring (SHM) plays a significant role for performance evaluation and condition assessments of the Nation's highway transportation assets, and it can promote the infrastructure 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], 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 and patched area), exposed rebar, cracking, abrasion (wear), and damage, etc. In this research, we introduce a data-driven visual inspection robot for spalling (with or without exposed rebar) and cracking inspection. The spalling and cracks are the main factors affecting the condition states of reinforcing concrete [3].

Automated visual inspection [4], [5], [6] has become a popular approach for structural surface inspection with the advance development of optics device technologies. 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 [5]. Under the support of FHWA LTBP program, an autonomous bridge deck inspection mobile robotic system was developed with visual cameras and other detection sensors [6]. Unmanned aerial vehicle (UAV) has also been deployed for bridge visual inspection [7]. Our wall-climbing robots provide vertical mobility to perform visual inspection and acoustic-based subsurface flaw inspection on both vertical and horizontal surfaces [8].

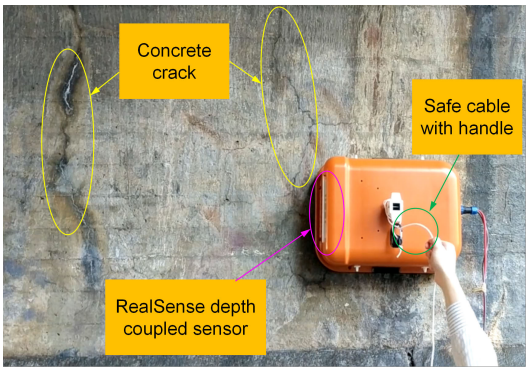


Fig. 1. Proposed wall-climbing inspection robot field test on the vertical surface of a bridge-tunnel at Riverside Dr & W 155th St, New York, NY 10032.

Various image processing algorithms have been explored for concrete structures surface crack and spalling inspection. As an earlier work, Oh et. al. [9] introduced 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. Crack-defragmentation approach of fragment grouping and fragment connection was proposed by Wu, and an artificial neural network (ANN) was used for crack detection classification [10]. More recently, convolutional neural network (CNN) has been deployed for crack classification on concrete structure images [11]. However, towards data-driven visual inspection of concrete structures, there are still some challenges needed to be solved. 1) high-

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,jxiao@ccny.cuny.edus

[2] Department of Automotive Engineering, Clemson University, South Carolina, USA bli4@clemson.edu

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

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

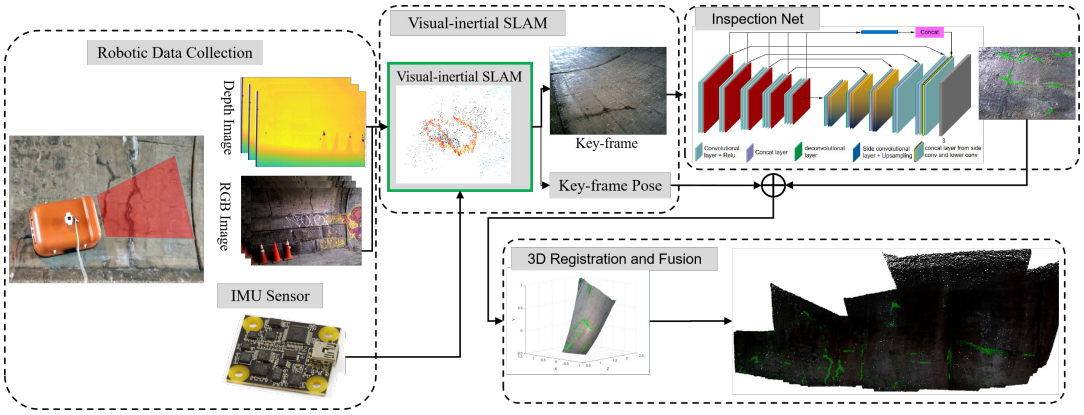


Fig. 2. The system architecture of our robotic metric inspection system. The whole system consists of four modules which are visual-inertial SLAM module, neural network inspection module, 3D frame registration module, and map fusion module.

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) 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, visual odometry positioning and 3D mapping software. This system aims at providing a holistic automated visual inspection data collection and analysis approach with semantic segmentation and 3D reconstruction 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 convolutional neural network for semantic segmentation based on our previous work [12].

II. System Architecture for Robotic Metric Inspection

This section describes the robotic metric inspection system architecture as illustrated in Fig.2, which consists of four modules: Robotic Data Collection, Visual-inertial SLAM (VI-SLAM), InspectionNet, 3D Registration and Map Fusion modules.

A wall climbing robot carrying an RGB-D camera and inertial measurement unit (IMU) sensor is used to collect RGB images and corresponding depth information of the concrete surface. Then, the VI-SLAM module takes RGB-D input and use ORB-SLAM [13] to obtain visual odometry and fuses with IMU measurement to perform real-time localization. The output of VI-SLAM module is a sequence of key-frames (RGB and depth images) and their corresponding pose estimations. To detect the surface flaws, we pass each key-frame RGB image through our InspectionNet and each pixel is classified into three categories (i.e. back-ground, crack and spalling) with probability prediction. The output is the class-aware images and can be used to calculate the metric measurement (i.e. length, width, and area) of the surface flaws (cracks and spalling). Finally, 2D to 3D registration and map fusion are performed to incrementally reconstruct the 3D map that highlights the surface flaws with different color and display their locations in 3D world coordination system for better and intuitive visualization.

A. Robotic Data Collection

We developed a wall-climbing robot to automate the data collect process for visual inspection of infrastructures (e.g., bridges, tunnels, dams and building facade). The robot uses an impeller and adjusts its speed to generate a negative pressure enclosed in a suction chamber and achieves a desired balance between strong adhesion force and high mobility [14]. The robot doesn't require perfect sealing and thus can move on both smooth and rough surfaces such as concrete wall, which is illustrated in Fig.2. The robot carries an on-board computer (i.e., Intel NUC computer), an Intel Realsense RGB-D camera and a Phidget IMU for vision-based inspection and a ground penetrating radar (GPR) for detection of subsurface objects (not covered in this paper). The Intel NUC computer connects RGB-D camera via USB port to collect the RGB and depth images and performs visual inertial SLAM in real time. It also streams the key-frame images through WIFI to a ground station computer with powerful CPU and GPU to perform image segmentation and 3D reconstruction.

B. Visual Inertial SLAM Module

This module takes the RGB and depth images to estimate the visual odometry of the robot at each frame using feature matching and optimization approaches proposed in ORB-SLAM [13]. It allows the odometry information to be updated at a rate of 30 Hz. We propose a new method to fuse the visual odometry with IMU measurements using a multi-state extended Kalman filter (MS-EKF) [15]. Thus the visual odometry can be updated in higher rate at 100 Hz to reduce the drift.

The steps are explained in detail as follows. For each RGB-D frame $\{I^{RGB}, I^{Depth}\}$, we developed a RGB-D visual odometry inspired by ORB-SLAM [13]. Meanwhile, the IMU measurements, which are acceleration (a_x, a_y, a_z) and angular velocity (w_x, w_y, w_z), are integrated to estimate the real time pose (R^{Imu}, t^{Imu}) of the robot as proposed in [16]. We regard the IMU acts as propagation and the camera as observation since IMU has a higher update rate. Once we obtain the estimated camera pose, we perform loosely-coupled fusion of the camera pose with the IMU pose to obtain the fused pose (R^f, t^f) that will reduce the motion drift [16]. We further perform local loop-closing check and bundle adjustment for adjacent frames to obtain the pose correction $C^{pose} = T(R^o, t^o)T^T(R^f, t^f)$, and update the MS-EKF state to further reduce the motion drift.

C. Neural Network Based Segmentation Model

Based on our previous work on semantic segmentation network [12], we introduce inspection neural network (InspectionNet) which is depicted in Fig.2. Inspired by VGG-16 [17], the architecture of InspectionNet has several improvements: 1) it adds five groups of deconvolutional layers to perform upsampling following the U-Net architecture [18]; 2) it introduces the side-layers which is proposed in Holistically-nested edge detection (HED) network [19].

The input to the InspectionNet is the key-frames generated from the VI-SLAM, which is a pair of RGB and depth images with associated pose. The InspectionNet performs class-aware segmentation on RGB images to classify each pixel into three classes (background, crack and spalling) with probabilistic prediction for each class label. The side-layers uses 3 channels output of each convolution group. The second to the last layer of the InspectionNet has a total of $6*3 = 18$ channels because we have a total 5 side layers plus the output from the deconvolutional layers. The last layer has 3 channels indicating the three classes (background, crack and spalling) with probabilistic prediction for each label. We implement the InspectionNet in Pytorch and deploy it in a GPU server for training and testing, which is discussed in Section.III.

D. 3D Registration and Map Fusion

The goal of our proposed robotic metric inspection system is not only to detect the existence of surface flaws (i.e., cracks and spalling) but also to find their physical location in 3D space and measure their property (length, width and area). In this paper, we introduce truncated signed distance function (TSDF) voxel volume [20] map to perform 2D to 3D reconstruction. For each RGB-D frame, given the camera intrinsic parameters K [21], we use the back-projection equation,

$$\begin{vmatrix} X = \frac{(u-u_0)}{f_x} \cdot I^{Depth}(u,v) \\ Y = \frac{(v-v_0)}{f_y} \cdot I^{Depth}(u,v) \\ Z = I^{Depth}(u,v) \end{vmatrix} \quad (1)$$

where $I^{Depth}(u,v)$ is the depth of the corresponding pixel (u,v) in the RGB image, and (X,Y,Z) is the 3D physical location corresponding to the pixel.

For the TSDF map, each RGB-D frame will fuse spatially by using surface reconstruction method proposed in [20] as follows: 1) perform ray-tracing based on the camera pose (R^o, t^o), and obtain the association from $3D$ voxel cell to $2D$ pixel; 2) calculate the ray angle θ to each voxel cell normal; 3) perform weighted fusion of the $3D$ cell position with the weight $W = cos(\theta)/I^{Depth}(u,v)$ and current 3D position which is obtained by using Equ.1.

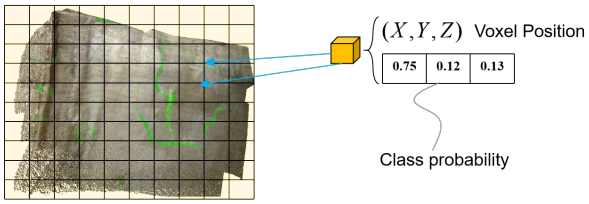


Fig. 3. The fusion of the inspection results is performed in 3D space using a probabilistic approach. For each voxel volume in the 3D map, it is a 3D cell with its center position and probability prediction for each class (background, crack and spalling).

For each voxel volume, it is a cubic cell with the center located at (X,Y,Z), and we also assign the class probability prediction to each cell, that is, each cell vv_i has a class probability for each class label, $P(vv_i) = \{p(vv_i)|p(j|vv_i) = c_j, i = 0,1,..,j = 0,1,2\}$ as depicted in Fig.3. Where $\sum_{j=0}^{j=2} p(j|vv_i) = 1$.

It is should be noted that the class probability of a 3D cell may vary when the camera view changes as robot moves, thus we introduce a Bayesian fusion method to perform class probability fusion of each 3D cell. Assuming we already know the 3D volume class probability prediction $P(vv_i) = \{p(vv_i)|p(j|vv_i) = c_j, i = 0,1,..,j = 0,1,2\}$ at time $k-1$. As camera view changes, we can obtain the 3D cell to 2D pixel mapping through ray tracing for new key-frame at time k. The class probability prediction of the corresponding pixel in the 2D image can be retrieved as $P(u,v) = \{p(u_i,v_i)|p(u_i,v_i,j) = c_j, i = 0,1,..,j = 0,1,2\}$. Then the class probabilistic prediction at each voxel volume can be updated through a recursive Bayesian filter,

$$P(vv_i)_k = P(vv_i)_{k-1} P(u_i,v_i)_k \quad (2)$$

where $P(p(vv_i)^m)_k$ denotes the class probability of voxel volume vv_i at time K, and $P(u_i,v_i)_k$ denotes the class probability of pixel (u_i,v_i) at time K.

After each voxel volume's class probability is updated, then we perform a max operation over the probabilistic vector to classify each voxel cell to one of the three categories,

$$P(vv_i) = max\{p(c_0|vv_i), p(c_1|vv_i), p(c_2|vv_i)\} \quad (3)$$

where c_0, c_1, c_2 denote the back-ground, crack, and spalling, $p(c_j|vv_i)$ denotes the probability of vv_i belong to c_j. The global 3D semantic map can be reconstructed incrementally using the 3D registration and fusion module that deals with the camera view changes.

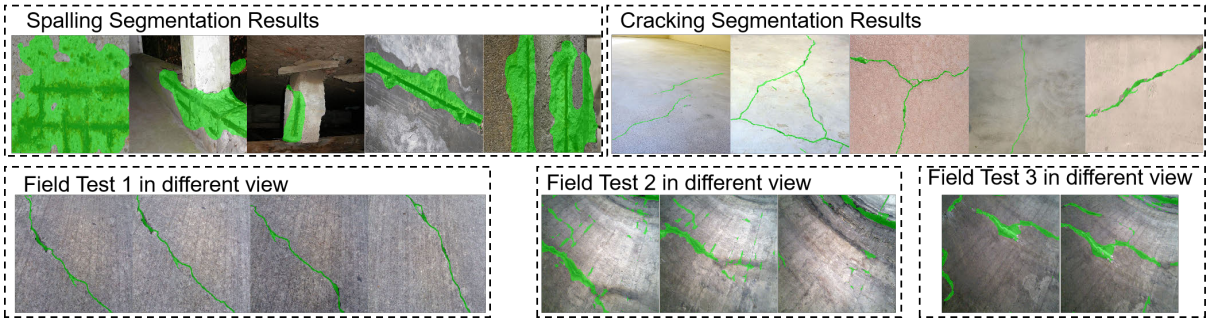


Fig. 4. The flaw segmentation demonstration on the test data set and the field test data. The field test 1 is performed by manually carrying the camera, and the field test 2 and 3 are performed by using the robot.

III. InspectionNet Training and Testing for Metric Inspection

This section discusses the data preparation and training of the InspectionNet, and we also discuses the metric fusion of label in 3D space using a probabilistic fusion approach to obtain the defects highlighted 3D map.

A. Data Preparation

To train the InspectionNet, we create a Concrete Structure Spalling and Crack (CSSC) dataset [11] and also use Berkeley Segmentation Dataset and Benchmarks 500 (BSDS-500) which is proposed by Arbelaez et. al. [22]. For BSDS-500, we directly use it to train the side layers based on edge detection task. For CSSC dataset, it has 298 spalling images and 512 crack images, which is not sufficient to train a large model like InspectionNet. Also, we noticed the following problems from our field tests,

- The crack and spalling appearance is highly affected by the distance from the robot camera to the surface.
- The surface flaw images suffer from low illumination which degrades the performance of the model.

To solve these problems, we introduce several approaches to augment the dataset. Firstly, considering the illumination problem, we use the Gamma Correction, $I(u,v) = 255 \times (\frac{I(u,v)}{255})^{\gamma}$ [23] to adjust the intensity of the RGB images in the data set. We adjust the Gamma values ranging from 0.25 to 2.0 with step of 0.25. Thus, we can have a total of 8 intensity different images to perform training. Secondly, we use random zoom (zoom in and out), elastic distortion, perspective transformation, size preserving rotation, and size preserving shearing which are proposed in [24], to augment the crack and spalling images and labels.

B. Model Training

The InspectionNet is to segment the pixels into 3 classes (i.e. back-ground, crack, spalling). Since spalling normally occupies a bigger area in an image and contains a larger number of pixels than crack, the InspectionNet model tends to overfit to spalling defects. Thus, we change the loss of the model as proposed in [18],

$$loss = \sum_{x_i \in X} w^j(x) log(p(x_{i,j})) \tag{4}$$

where x_i indicate a pixel given image X, $p(x_{i,j})$ is the pixel x_i probabilistic prediction over class j, and w^j is the weight of each classes. In this paper, we will discuss whether the weight affect the segmentation result, and how much does the weight affect the segmentation result.

To train the InspectionNet, we split the whole training into two procedures: 1) we train the HED [19] model and have the side layer output 3 channels using BSDS-500 dataset. We train the HED model in 100 epochs. 2) we use the HED side layers' weight to initialize the side layers of InspectionNet and use VGG-16 pre-trained model to initialize the left side convolutional kernel of the InspectionNet. The deconvolutional and the right side convolutional layers are randomly initialized. For all the layers, the parameters can be updated.

C. 3D Metric Measurement

Our goal is to recognize the metric measurement of each surface flaw patch and determine its location in 3D space. For each key-frame image, we classify each pixel into one of the three classes and find the flaw patch (crack or spalling) and the depth information using the InspectionNet. Then, the 3D registration and map fusion module register each pixel back to the 3D space to obtain the physical location of the corresponding pixel. We further generate a mesh based on the 3D information of the surface flaw patch [25]. Finally, we detect and characterize the flaw patch by calculating the width, height, and area information based on the mesh.

IV. EXPERIMENTAL STUDY

To evaluate our approach, we perform 6 tests which include CSSC dataset test, two field tests using hand-held camera, and three automated field tests using the wall-climbing robot. The field tests of robotic inspection system were conducted on a 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 robotic inspection system is shown in video demo video [1].

A. InspectionNet Training and Evaluation

Dataset Based on the CSSC dataset, in which 298 spalling images and 954 crack image, among them 522 are labeled

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

TABLE I

INSPECTIONNET WEIGHT EFFECTS. *E_MaxF1* IS EVALUATION MAXF1 SCORE, *E_AP* IS EVALUATION AVERAGE PRECISION, *T_MaxF1* IS TRAINING MAXF1 SCORE, *T_AP* IS TRAINING AVERAGE PRECISION, *T_BAP* IS TRAINING CONCURRENT PRECISION, *AF* IS THE AVERAGE FREQUENCY.

	InspectionNet											
	Weight of classes			Weight of classes			Weight of classes			Weight of classes		
	BG	Crack	Spalling	BG	Crack	Spalling	BG	Crack	Spalling	BG	Crack	Spalling
	0.2	4	1	0.2	4	4	0.2	8	4	0.2	4	8
E_ MaxF1	53.8389			23.4860			54.6388			46.0032		
E_ AP	50.0527			13.7970			36.0459			28.7359		
T_ MaxF1	58.5810			43.4980			37.8984			31.5302		
T _ AP	55.3143			30.0418			20.2461			18.5447		
T_ BAP	86			80			0.74			0.75		
AF	6.1628			6.8887			6.9698			6.8661		

images. We first perform data augmentation as explained in Section III-A, and obtain a total of $58,149$ images for the crack model, where $40,911$ images for training, $6,474$ for cross-validation, and $10,764$ for testing. The spalling detection model has $8,151$ for training, $1,170$ for cross-validation, and $2,210$ for testing.

Neural Network Training

We train the InspectionNet on a GTX 1080 GPU server computer and use Pytorch to deploy the algorithm. For the InspectionNet, we use the stochastic gradient decent (SGD), with an initial learning rate at 0.001, momentum as 0.9, and weight decay of 5×10^{-5}. The model is set to run in a $12,000$ iterations and lasting around half day. To compare the performance between models, we use metrics such as batch concurrent accuracy, average precision, and max F1 score.

Accuracy Evaluation We first perform the crack and spalling inspection validation using the test data set from the CSSC data set. For testing purpose, we use total of $10,764$ crack images, and $2,210$ spalling images.

The comparative accuracy performance under different weight setting of the InspectionNet is given in Table.I, where we provide a comparison under 4 different weight settings. Before performing the weight comparison, we first calculate the number of all the crack, spalling, and background (BG) pixels, and find the distribution of the three classes are $(20,1,4)$. It is illustrated in Table.I, the four settings are $(0.2,4,1)$, $(0.2,4,4)$, $(0.2,8,4)$, and $(0.2,4,8)$. We can conclude that the inverse weight over the three classes number distribution, that is, $(0.2,4,1)$ allows the model to perform an average 10% higher in average accuracy compared with other settings.

Besides, we also conduct experiments to compare VGG-Unet and the FCN-8s. We also compared with FCN-32s, and we found that FCN-32 is not able to perform segmentation on crack images or segmentation on tiny spalling flaws. For average precision, the InspectionNet can obtain 83.58% which is almost 3% higher than VGG-Unet. For FCN-8s, same as FCN-32s, is not able to perform crack segmentation.

Processing Speed

To evaluate the processing speed, we perform 5 sets of testing, including 3 for crack detection and 2 for spalling detection. We calculate the mean processing speed of each session. Based on fives test set, we found that our InspectionNet has average 6.2 frames per second which is sufficient for online processing.

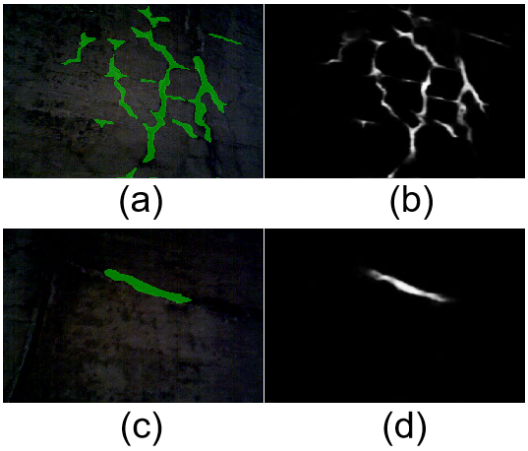


Fig. 5. An extreme demonstration of the segmentation model under low illumination environment. (a) and (c) shows the green color overlaid results, and (b) and (d) are the probabilistic distribution of spalling over the original image.

B. Dataset Testing and Field Test

The evaluation of the visual inspection system is performed in two steps. First, we test the detection performance on the test dataset and quantitatively evaluate the average accuracy. In the second step, we perform field tests under a bridge with 3D reconstruction. In the field tests, we consider both normal illumination and low illumination situation to perform inspection and 3D reconstruction.

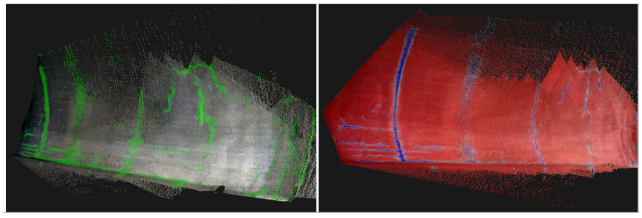
The performance of detection on CSSC dataset is illustrated in Fig.4. In this image, we overlay the spalling and crack flaws using green color. In spalling segmentation results, we show that the model performs a robust inspection under any visual scale. For the crack inspection, we can see that the most left image has a huge view-variance compare to the normal data, and the InspectionNet can still segment the crack out. We also provide the inspection results of three field tests. For each field test, we can seen that our model is robust to surface contrast and the view changes.

Illumination Robustness

Besides the scale and view robustness, we also test the model robustness under low illuminated environment. We perform two set of experiments under the bridge. The result are illustrated in Fig.5. For image (b) and (d), they are the class probability prediction, and the white indicate the high probability region where has flaws and the black denotes the back-ground. Image (a) and (c) are the green color overlaid images for better visualization. It can conclude that our InspectionNet is robust to low intensity images.

3D Metric Semantic Registration

Our goal is to perform metric semantic reconstruction as illustrated in Fig.6. The 3D reconstruction is performed by coupling the image frames with *pose* and *time*, where the frames are key-frames from VI-SLAM. Then, the InspectionNet performs the flaw segmentation on the RGB image. Thus we can register the key-frame to the 3D space with semantic information. It is illustrated in Fig.6, the green area (Fig.6.(a)) and blue area (Fig.6.(b)) denote the flaw patches, where all the information are with metric scale. Then civil engineers can determine the location of each flaw patch and calculate its metric property (width, length, and area).



(a) Defects highlight over original color (b) Heat map representation of the concrete structure

Fig. 6. 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.

V. CONCLUSION

This paper introduces a cutting-edge deep learning-based visual inspection system. The system leverages visual-inertial SLAM positioning and 3D reconstruction, and uses InspectionNet for segmentation, the flaw patches are registered in the 3D model to provide metric information for concrete structure condition assessment. The concrete structure spalling and crack (CSSC) dataset and InspectionNet are released as source code to the research communities. The field experiments show the effectiveness of our proposed metric inspection methodology in field test scenarios.

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] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] L. Yang, B. Li, W. Li, B. Jiang, and J. Xiao, "Semantic metric 3d reconstruction for concrete inspection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1543–1551.

[13] 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.

[14] L. Yang, G. Yang, Z. Liu, Y. Chang, B. Jiang, Y. Awad, and J. Xiao, "Wall-climbing robot for visual and gpr inspection," in *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*. IEEE, 2018, pp. 1004–1009.

[15] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint kalman filter for vision-aided inertial navigation," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 2007, pp. 3565–3572.

[16] L. Armesto, J. Tornero, and M. Vincze, "Fast ego-motion estimation with multi-rate fusion of inertial and vision," *The International Journal of Robotics Research*, vol. 26, no. 6, pp. 577–589, 2007.

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

[18] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," pp. 234–241, 2015.

[19] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1395–1403.

[20] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *2011 IEEE International Symposium on Mixed and Augmented Reality*. IEEE, 2011, pp. 127–136.

[21] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, 2015.

[22] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.

[23] C. Poynton, *Digital video and HD: Algorithms and Interfaces*. Elsevier, 2012.

[24] M. D. Bloice, C. Stocker, and A. Holzinger, "Augmentor: an image augmentation library for machine learning," *arXiv preprint arXiv:1708.04680*, 2017.

[25] X. Liu, N. Qin, and H. Xia, "Fast dynamic grid deformation based on delaunay graph mapping," *Journal of Computational Physics*, vol. 211, no. 2, pp. 405–423, 2006.