13 May 2020

Modeling and Simulation of a Robotic Bridge Inspection System

Md Monirul Karim

Cihan H. Dagli
Missouri University of Science and Technology, dagli@mst.edu

Ruwen Qin
Missouri University of Science and Technology, qinr@mst.edu

Recommended Citation

M. M. Karim et al., "Modeling and Simulation of a Robotic Bridge Inspection System," *Procedia Computer Science*, vol. 168, pp. 177-185, Elsevier B.V., May 2020.
The definitive version is available at https://doi.org/10.1016/j.procs.2020.02.276

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 168 (2020) 177–185

Procedia Computer Science

www.elsevier.com/locate/procedia

Complex Adaptive Systems Conference with Theme:
Leveraging AI and Machine Learning for Societal Challenges, CAS 2019

Modeling and Simulation of a Robotic Bridge Inspection System

Muhammad Monjurul Karim[a]*, Cihan H. Dagli[a], Ruwen Qin[a]

[a]Department of Engineering Management and Systems Engineering, Missouri Unversity of Science and Technology, Rolla, MO 65409, USA

Abstract

Inspection and preservation of the aging bridges to extend their service life has been recognized as one of the important tasks of the State Departments of Transportation. Yet manual inspection procedure is not efficient to determine the safety status of the bridges in order to facilitate the implementation of appropriate maintenance. In this paper, a complex model involving a remotely controlled robotic platform is proposed to inspect the safety status of the bridges which will eliminate labor-intensive inspection. Mobile cameras from unmanned airborne vehicles (UAV) are used to collect bridge inspection data in order to record the periodic changes of bridge components. All the UAVs are controlled via a control station and continuously feed image data to a deep learning-based detection algorithm to analyze the data to detect critical structural components. A cellular automata-based pattern recognition algorithm is used to find the pattern of structural damage. A simulation model is developed to validate the proposed method by knowing the frequency and time required for each task involved in bridge inspection and maintenance. The effectiveness of the model is demonstrated by simulating the bridge inspection and maintenance with the proposed model for five years in AnyLogic. The simulated result shows around 80% of man-hour can be saved with the proposed approach.

Peer-review under responsibility of the scientific committee of the Complex Adaptive Systems Conference with Theme: Leveraging AI and Machine Learning for Societal Challenges

Keywords: Simulation modeling; bridge inspection; UAV, cellular automata, deep learning, discrete-event simulation, AnyLogic

1. Introduction

The U.S transportation system has more than 600,000 bridges, average age of these bridges are 42 years, however, most of these exceed the lifetime they were built to have [1]. As a means for transportation, hundreds of thousands of civilians use bridges every day. According to regulation, each bridge requires inspection every two years to ensure safety for the civilians [2]. This means that every month, around 25,000 bridges need to be inspected. The current bridge inspection process is manual, involving visual inspection with heavy lifting equipment and requires people to work from a dangerous height. Moreover, it requires the closure of the

* Corresponding author. Tel.: +1-609-787-9233
E-mail address: mk5n2@mst.edu

1877-0509
Peer-review under responsibility of the scientific committee of the Complex Adaptive Systems Conference with Theme: Leveraging AI and Machine Learning for Societal Challenges
10.1016/j.procs.2020.02.276

road during the time of inspection causing traffic congestions. The average inspection cost per bridge ranges from $4,500-$10,000. These make the bridge inspection operation one of the most costly operations in the state Department of Transportation [3]. To address this issue, a remotely controlled robotic platform is required to inspect the safety status of the bridges that will eliminate labor-intensive inspection and allow the bridges to be visually assessed from a remote location.

Recent years have witnessed the rising of research interests in bridge inspection methodology [4], [5], [6]. To automate the inspection process, researchers have proposed many methods. For example, laser scanning method has been developed for data collecting [7], [8], [9]. This technique uses pulse of laser light to acquire geometric data for bridge inspection. However, this approach requires heavy laser equipment that is very expensive. Besides, success of this method is largely dependent upon the diligence and education of inspection workers. Therefore, researchers have started developing robotic system for inspecting the safety status of bridges. For example, Oh et al. [10] proposed a robotic system that involves a specially designed car, a robot mechanism, and control system to gather crack data from the bridge using computer vision. Tung et al. [13] developed a mobile manipulator imaging system for the automation of bridge crack inspection. This approach requires two charge coupled cameras on a mobile vehicle to collect bridge images. Most of these robotic based approaches require a ground vehicle to carry the camera that also causes the closure of the road. Besides, in a bridge, there are many places that are inaccessible by ground vehicles. Therefore, we propose a remotely controlled robotic platform using a mobile camera from an unmanned airborne vehicle (UAV) to collect bridge image data.

On the other hand, to analyze the data researchers have studied various approach to find out the cracks in the bridges from the image data. For example, Sohn et al. [14] monitored crack changes in the concrete structure. They focused on quantifying the periodic change in the cracks from multi-temporal images. Ito et al. [15] presented a system to inspect concrete block by means of analyzing fine crack extraction. All these approaches only detect cracks of a certain type. These approaches cannot be used for detecting multiple types of damages in the bridges. To address these drawbacks, recently deep learning based approaches are thoroughly studied to determine the damage in structure. Karim et al. [16] developed a two-staged threshold based object detection method that can detect multiple objects in an image. They used Mask R-CNN [17] based object detector. However, these approaches just only detect cracks in the bridges at the exact time of inspection. These approaches cannot detect a pattern of crack propagation from the images. Besides, all the approaches are problem specific and are studied as separate problems. Therefore, a complex system having the capability of solving all these separate problems as a single problem is greatly in need. Motivated by this need, in this paper a model has been developed combining bridge data collection, data processing, and data analyzing system together to have a complex system to efficiently inspect structural health condition.

The system uses a region based deep learning algorithm to accurately detect and segment cracks in the images of structural components. Then a cellular automata based pattern recognition algorithm has been used to get the pattern of crack propagation in the structural component. For pattern recognition, 5 rules have been established to simulate real-world crack propagation in bridges. Bridge experts can take the maintenance decision from the crack propagation rate of the bridges. To validate the proposed model, a simulation model has also been developed in this study to simulate the proposed model for five years in order to determine the frequency and time required for complete bridge inspection and maintenance. This simulation model can work as a decision support tool for taking maintenance decision by the decision makers.

The remainder of this paper is organized as follow: Section 2 delineates the conceptual model for bridge crack detection and segmentation using UAV, followed by examples illustrating the implementation of the method. Numerical simulation and discussion from the examples are illustrated in Section 4. Conclusions and future work are summarized at the end, in Section 5.

2. Methodology

The proposed model for bridge inspection with UAV is illustrated in Fig. 1. In a certain region, a robot such as a UAV takes videos of a bridge during the inspection. After completing the inspection, the video data are converted into image data. Each image then passes through a deep learning segmentation tool frame by frame. The segmentation tool is pre-trained on a large dataset. This segmentation tool detects and segments cracks of the bridge in images. Images not containing any crack information are discarded from the pool of image frames. Then heatmaps of detected images are generated. These heatmaps of cracked images are given as input to the cellular automata. The rules of the cellular automata then determine the crack propagation rate. Based on the crack propagation rate, decision makers make maintenance decision.

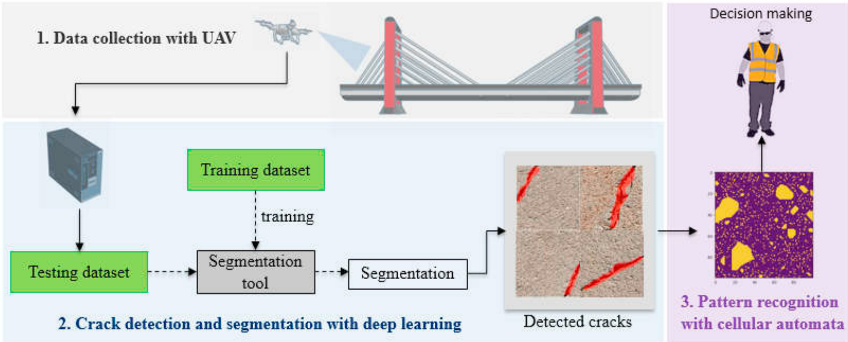


Fig. 1. Proposed model for bridge inspection with unmanned airborne vehicle.

2.1. Deep learning algorithm

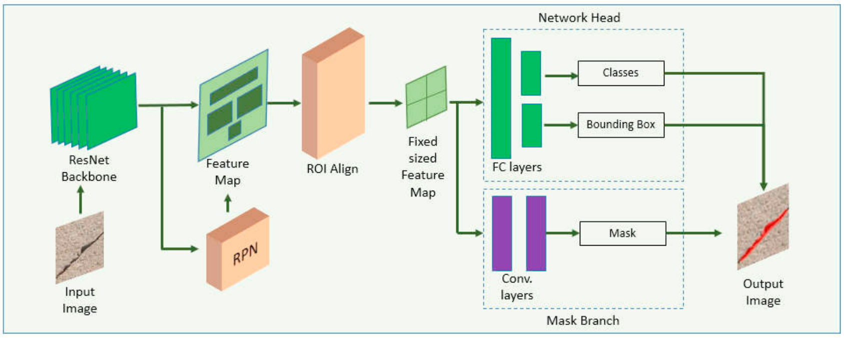


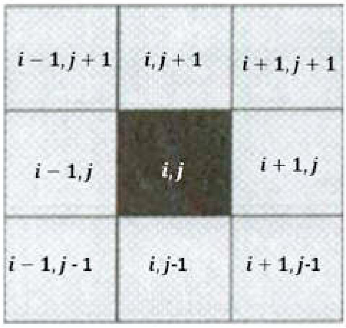
Fig. 2. Architecture of the Mask R-CNN

Mask R-CNN is a region based convolutional neural network, can effectively detect and segment detected objects at the pixel level. In this study, Mask R-CNN has been chosen as the segmentation tool for detecting and localizing cracks in the structural components. Fig.2 illustrates the structure of this algorithm. This algorithm has a ResNet [19] based Feature Pyramid Network (FPN) which works as the feature extractor to generate feature map from the input image. Then a Region Proposal Network (RPN) is applied to the feature maps. An RPN is a neural network that slides over the image to create possible proposal boxes which are called anchors. These anchors are ranked to find the top anchors that are likely to contain objects. These are called the Region of Interests (RoI). Then these RoIs are aligned with the input image and converted into fixed size feature maps by a layer called Region of Interests Align (RoIAlign). These fixed-size feature maps are passed through two independent branches: network head branch to perform object classification and bounding box generation, and the mask branch to generate instance masks on top of the detected objects. Interested readers can refer to [17] for details.

In this study, a trained Mask R-CNN takes all the input images and detects all possible cracks in the images.

2.2. Cellular Automata

After detecting the possible cracks in an image a cellular automata based pattern recognition algorithm is applied to the images. The purpose of this cellular automata is to simulate the crack propagation in the bridge structure. Based on the simulation result, crack propagation rate can be determined. An image containing cracks can be considered as a lattice space of many cells. The idea of cellular automata is that the behavior of each cell is dependent on the behavior of the neighboring cells. For example, if a cell without crack is surrounded by many cracked cells, it is highly likely that the crack will be propagated to the cell without crack. Let's consider a cellular automaton consists of a regular lattice of sites. Each site takes on k possible values, and is updated in discrete time steps according to a rule ϕ that depends on the value of sites in some neighborhood around it. Fig. 3 shows a neighborhood structures considered for two-dimensional cellular automata. For this study, number of rules $\emptyset = 5$. Based on these 5 rules, cells are updated in discrete time steps.



- **Rule 1:** If the target grid (i, j) is not cracked and more than 3 neighbors are cracked then the crack will propagate to the target grid
- **Rule 2:** If three consecutive frames $(i-1, j+1), (i, j+1)$ *and* $(i+1, j+1)$ are cracked then the crack will propagate to the target grid (i, j)
- **Rule 3:** If three consecutive frames $(i-1, j+1), (i-1, j)$ *and* $(i-1, j-1)$ are cracked then the crack will propagate to the target grid (i, j)
- **Rule 4:** If three consecutive frames $(i+1, j+1), (i+1, j)$ *and* $(i+1, j-1)$ are cracked then the crack will propagate to the target grid (i, j)
- **Rule 5:** If three consecutive frames $(i-1, j-1), (i, j-1)$ *and* $(i+1, j-1)$ are cracked then the crack will propagate to the target grid (i, j)

Fig. 3. Cells location in a lattice space and the rules associated in each cell. Here, (i, j) is the location of the target cell. Which is surrounded by 8 neighbors.

2.3. Simulation

The developed model in this study is a discrete event model. Which is simulated using a proprietary simulation software namely Anylogic. In the Anylogic, the discrete event model is specified graphically as a process flowchart where blocks represent operations. The flowchart starts with source that generates agents and inject them into the process and ends with sink blocks that remove them. The paper describes the development of a simulation model for bridge inspection with UAV with time windows within AnyLogic simulation environment. The defined agents for the simulation are UAVs, deep learning machines and maintenance team.

3. Application of the proposed model

Real world bridge image data has been used in this study as the starting point. For bridge data collection a mobile camera attached with a multicopter UAV has been used. The camera captured videos of two bridges (bridge 1 and bridge 2) at two different locations. The average speed (v) of the UAV was 20 mph. The framerate of the captured video is 30fps with 3840 x 2160 resolution. A testing dataset (D) has been created with 4672 images from bridge 1 and 2. The segmentation tool is fine-tuned with a training dataset (T_0) of 1500 images containing cracks.

3.1. Model training and fine-tuning

ResNet feature extractor of the segmentation tool was initialized with weights pre-trained on the Microsoft COCO dataset. The model was fine-tuned using T_0. At first, the network head was trained for 30 epochs and all the parameters in the previous layers were fixed. Then, the ResNet Backbone C5 and the network head were trained for 100 additional epochs, and all other layers were fixed. Each epoch consists of 100 training iterations. Stochastic gradient descent was used as the optimizer and the momentum was 0.9. The learning rate was 0.001 for the first 30 epochs of training, and it is reduced to 0.0001 for the remainder 100 epochs of training. The batch size of one image was used on a single NVIDIA Geforce GTX 1080Ti GPU for this training process that took about 22 hours to complete.

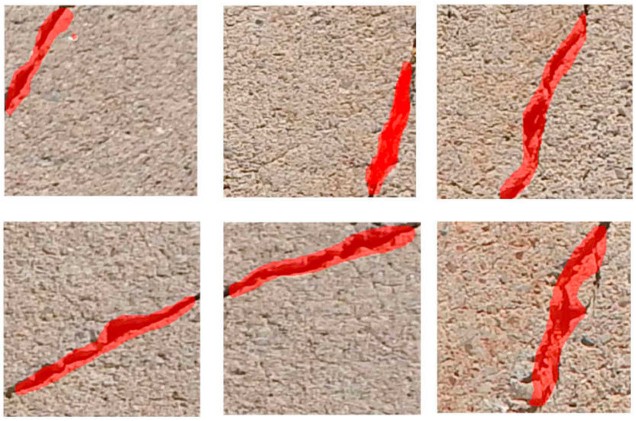


Fig. 4. Examples of detected cracks with the deep learning algorithm

3.2. Crack detection

The trained segmentation tool is tested on the dataset D. The tool successfully detected and segmented the cracks in the concrete structure. Fig. 4 shows some successful examples of the detected cracks. The red masks indicate the segmented cracks. The masks tightly overlapped with the corresponding cracks. No obvious false positives were found in those examples. From the figure, it can be seen that, cracks position and pattern are random. However, the detector detected the cracks from a different angle. Moreover, there was motion blur because of the UAV motion. This motion blur may somehow affect the detection result in some frames. However, those undetected frames can be ignored from the consideration. As the frame rate of the camera was 30fps. That means many frames are almost identical to each other. Therefore, few of the identical frames can obviously be ignored and will not hamper the overall result. The detected images are used for giving input to the cellular automata.

3.3. Identifying crack propagation rate with cellular automata

A cellular automata has been simulated to determine the crack propagation rate of the bridge based on the rule of crack propagation described in section 2.3. To simplify the simulation, we initiated the simulation with a lattice space of size 100 x 100. Simulation is initiated by the heat map of an image containing crack. Initial probability for each cell in the region other than the cracked region of being cracked is considered 0.1 and not being cracked is considered 0.9. Update interval for each iteration is considered 100ms. At each iteration, crack will be propagated based on the five rules set for the simulation. The lattice space contains 10,000 cells. Total number of iteration (I) required for completely turning all these cells into cracked cells is calculated. The higher iteration required for this complete transition means the crack propagation rate is slow. For N cells and I iteration, the crack propagation rate, $r = {}^{I}/_{N}$. In the Fig. 5, three iterations (at three different time steps) are shown. The yellow cracks get propagated at each time interval. Observing the crack propagation rate, decision makers can take decision when and what part of the bridge will require maintenance.

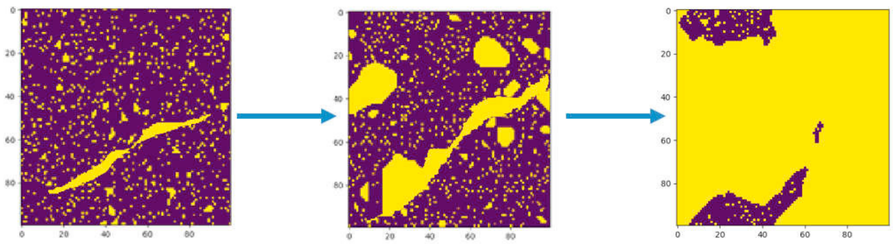


Fig. 5. Crack propagation at 3 time steps. The yellow cells represent cracked cells. At each update interval, it gets propagated.

4. Numerical simulation and discussion

To validate the proposed model a numerical simulation is performed using discrete-event simulation of AnyLogic software. The whole model is simulated for 5 years in two bridges of two different size. Length of bridge 1 and 2 are 900 meters and 700 meters respectively. Deep learning machine can process 2 images per second.

Hour is considered as the unit time for the simulation. Three main agents have been considered, which are UAV, deep learning machine and maintenance team. For simulation individual logic has been developed for each agent.

4.1. Logic for UAVs

Two different process flow diagram has been developed for inspecting two different bridge. Fig. 6 indicates the logic for both the bridge inspection by the UAV. The upper logic in the figure represents the logic for bridge 1 and the lower logic of the figure represents the logic for the bridge 2. Source nodes generate UAV. timeMeasureStart function takes the start time of the bridge inspection. Then next two moveTo functions determine the movement of the UAVs in the predefined path in Anylogic. Range of the UAV speed is set 15 to 20mph. TimeMeasureEnd() function calculates cycle time required for bridge inspection. Sink nodes remove the UAV from the process flow. From the figure, it can be seen that for the first bridge in 5 years there will be a total of 89 cycles of inspection and for bridge 2, 81 cycles of inspection.

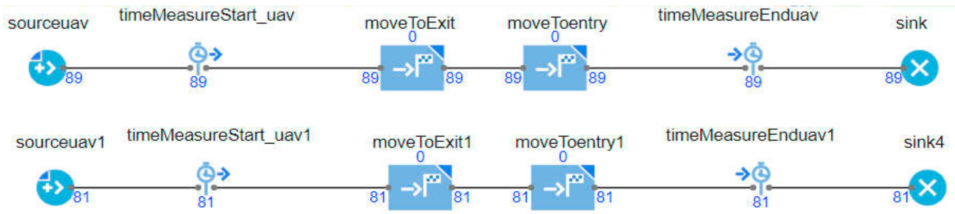


Fig. 6. Logic for UAV operation

4.2. Deep learning logic and decision making

After completing one cycle of inspection, information is generated and is transferred to the deep learning machine. Deep learning machine processes the data. The machine can perform image processing and pattern recognition at a speed of 2fps. The processing of data is represented by the delay() functions (dl_delay, dl_delay1) in the process flow of the AnyLogic. After processing the data again information will be generated which will again be processed for decision making. The delay for decision making is represented as decision_delay() functions. Similar to UAV, here also time is measured with timeMeasure() functions.

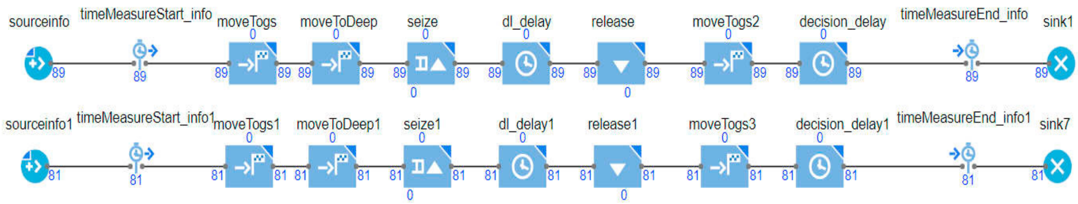


Fig. 7. Logic for deep learning and decision making

4.3. Logic for maintenance

Based on the crack propagation rate, the maintenance team will decide to go for maintenance if required. Here, the selectOutput() function determines the probability of maintenance requirement. This probability of maintenance required is determined by the bridge expert based on the crack propagation rate generated by the cellular automata. For simplicity in this simulation, the probability of doing maintenance is considered 0.7 for bridge 1 and 0.6 for bridge 2. From the process flow diagram, it is visible that for bridge 1, maintenance was required for 61 times out of 89 times and for bridge 2, it was required for 52 times out of 81 times of inspection.

If crack propagation rate is low, it can be assumed that bridge maintenance is not required as the bridge is in good condition. Hence, for a certain period of time bridge inspection is not necessary as it also can be assumed that there will not be any sudden deterioration in the bridge. In this study, we assumed the duration of this period is two months and represented by the delay() function in the process flow diagram. The red bounding boxes in the process flow diagram indicates the process for not doing maintenance.

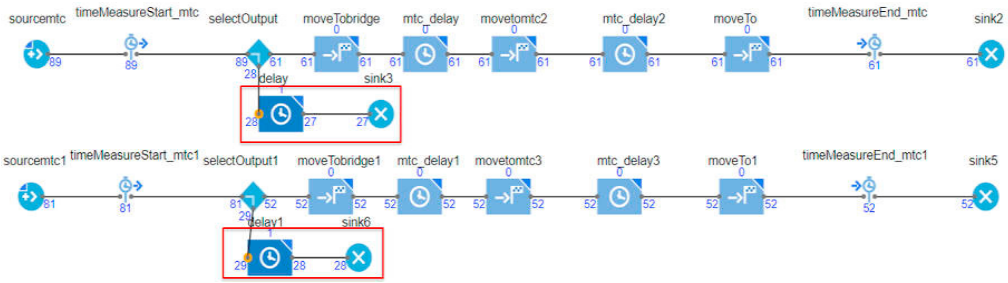


Fig. 8. Logic for maintenance

4.4. Statistics report

After simulating the model for five years, a statistical report has been generated in the AnyLogic. Fig. 9. (a) represents the cycle time required for UAV inspection. UAV1 represents the inspection for bridge 1 and Uav2 represents inspection for bridge 2. Fig. 9(b) represents the histogram of the time required for each cycle. From the histogram, it is visible that, the mean time required for each cycle of inspection is 0.54 hour and 0.47 hour respectively for bridge 1 and 2.

a

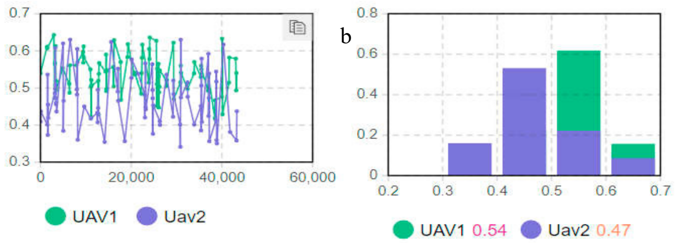


Fig. 9. (a) cycle time of inspection (b) histogram of cycle times

Fig. 10. (a) Represents the cycle time required for deep learning machine and decision making. DeepLearning_bridge1 represents the data processing time required for bridge 1 and DeepLearning_bridge2 represents data processing time required for bridge 2. Fig. 10(b) represents the histogram of the time required for each cycle. From the histogram, it is visible that, the mean time required for each cycle is 9.05 hours and 7.54 hours for bridge 1 and 2 respectively.

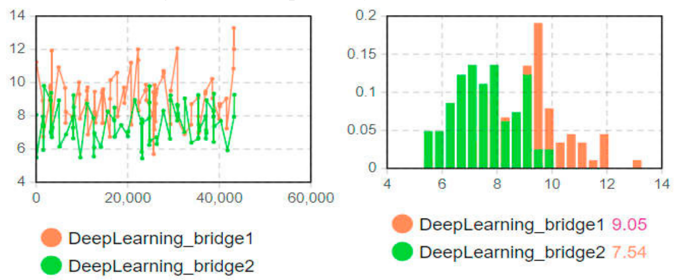


Fig. 10. (a) Cycle time for deep learning and decision making (b) histogram of cycle times

Fig. 11. (a) Represents the cycle time required for maintenance. Mtc_bridge1 represents the time required for bridge 1 and Mtc_bridge2 represents the time required for bridge 2. Fig. 11(b) represents the histogram of the time required for each cycle. From the histogram, it is visible that, the mean time required for each cycle is 57.18 hours and 44.74 hours respectively for bridge 1 and 2.

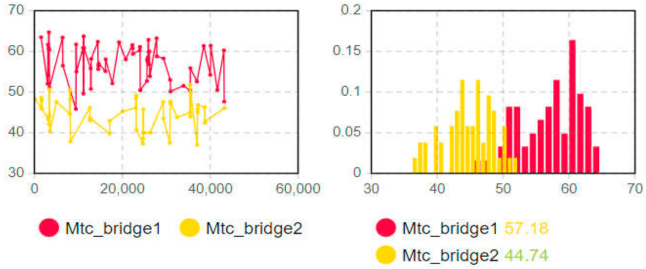


Fig. 11. (a) Cycle time for maintenance (b) histogram of cycle times

Table 1 illustrates the frequency and time required by the agents in 5 years' time period. From the table, it can be seen that bridge 1 takes higher time for all three activities (i.e. inspection, deep learning, and maintenance) than bridge 2.

Table 1. Frequency and time required by the agents in five years

		Agents	Frequency	Time (hour)	Total Time (hour)
Bridge 1	Inspection & Analysis	UAV inspection	89	48	854
		Deep learning used	89	805	
	Maintenance	Maintenance required	61	3,488	3,488
Bridge 2	Inspection & Analysis	UAV inspection	81	38	648
		Deep learning used	81	610	
	Maintenance	Maintenance required	52	2,326	2,326

It can be observed from the simulated result, total of 854 hours for bridge 1 and 648 hours for bridge 2 will be required for inspecting and analyzing the data in the five years of period. On the other hand, traditional manual bridge inspection requires 24 man-hours to 48 man-hours for a bridge of 1000 meter length [20]. Therefore, the traditional method may require 2,136 man-hours to 4,272 man-hours to complete inspecting bridge 1 in five years. This signifies that UAV based bridge inspection method can save 60% to 80% of the inspection time.

5. Conclusion

This paper presents a vision sensor-based system that monitors and inspects bridges to detect and locate cracks in the bridges. This ability enables the state department of transport to know the exact location of a crack in the structural component. After detecting the location of the cracks, a cellular automata based pattern recognition algorithm determines the crack propagation rate. Based on this rate, decision-makers can easily make a maintenance schedule. This paper also presents an AnyLogic based simulation model to validate the proposed method. This simulation model could be used as a decision support tool for advanced analysis of the bridge inspection and maintenance schedule. Results of the simulation model are promising enough to be useful in a real-world scenario.

In this study, real bridge image data are used to locate the cracks in the structure. However, instead of using all the images as the input of cellular automata, some sample images were used. Besides, 5 rules are assumed to simulate crack propagation. Our future work will focus on performing a complete case study using all the images to validate the assumptions and make a comparative study between the simulated result and real observational result.

Acknowledgements

Financial support for this INSPIRE UTC project is provided by the U.S. Department of Transportation, Office of the Assistant Secretary for Research and Technology (USDOT/OST-R) under Grant No. 69A3551747126 through INSPIRE University Transportation Center (http://inspire-utc.mst.edu) at Missouri University of Science and Technology. The views, opinions, findings and conclusions reflected in this publication are solely those of the authors and do not represent the official policy or position of the USDOT/OST-R, or any State or other entity.

References

[1] ASCE (2017). "2017 infrastructure report card." Reston, VA: ASCE

[2] “Highway bridge inspections,” Jun 2017. [Online]. Available: https://www.transportation.gov/content/highway-bridge-inspections

[3] Zulfiqar, Ahsan, et al. (2015) "Design of a bridge vibration monitoring system (BVMS)." 2015 Systems and Information Engineering Design Symposium. IEEE.

[4] Thompson, Paul D., and Richard W. Shepard. (2000) "AASHTO Commonly-recognized bridge elements." Materials for National Workshop on Commonly Recognized Measures for Maintenance, Scottsdale, Arizona.

[5] Jáuregui, David V., and Kenneth R. White. (2003) "Implementation of virtual reality in routine bridge inspection." Transportation research record 1827(1): 29-35.

[6] Bauer, Johannes, Niko Sünderhauf, and Peter Protzel. (2007) "Comparing several implementations of two recently published feature detectors." IFAC Proceedings Volumes 40 (15): 143-148.

[7] Ditto, Thomas D., Jim Knapp, and Shoshana Biro. (2009) "3D inspection microscope using holographic primary objective." Optical Inspection and Metrology for Non-Optics Industries. Vol. 7432. International Society for Optics and Photonics.

[8] Sansoni, Giovanna, Marco Trebeschi, and Franco Docchio. (2009) "State-of-the-art and applications of 3D imaging sensors in industry, cultural heritage, medicine, and criminal investigation." Sensors 9 (1): 568-601.

[9] Jaselskis, Edward J., Zhili Gao, and Russell C. Walters. (2005) "Improving transportation projects using laser scanning." Journal of construction engineering and management 131 (3): 377-384.

[10] Oh, Je-Keun, et al. (2009) "Bridge inspection robot system with machine vision." Automation in Construction 18 (7): 929-941.

[11] Lim, Ronny Salim, et al. (2011) "Developing a crack inspection robot for bridge maintenance." 2011 IEEE International Conference on Robotics and Automation. IEEE.

[12] Yu, Seung-Nam, Jae-Ho Jang, and Chang-Soo Han. (2007) "Auto inspection system using a mobile robot for detecting concrete cracks in a tunnel." Automation in Construction 16(3): 255-261.

[13] Tung, Pi-Cheng, Yean-Ren Hwang, and Ming-Chang Wu. (2002) "The development of a mobile manipulator imaging system for bridge crack inspection." Automation in construction 11 (6): 717-729.

[14] Sohn, Hong-Gyoo, et al. (2005) "Monitoring crack changes in concrete structures." Computer-Aided Civil and Infrastructure Engineering 20 (1): 52-61.

[15] Ito, Atsushi, Yoshimitsu Aoki, and Shuji Hashimoto. (2002) "Accurate extraction and measurement of fine cracks from concrete block surface image." IEEE 2002 28th Annual Conference of the Industrial Electronics Society. IECON 02. Vol. 3. IEEE.

[16] Karim, Muhammad Monjurul, et al. (2019) “A Region-Based deep learning algorithm for detecting and tracking objects in manufacturing plants,” in 25th International Conference on Production Research 2019 (ICPR 2019), Chicago, USA.

[17] He, Kaiming, et al. (2017) "Mask r-cnn." Proceedings of the IEEE international conference on computer vision.

[18] Makantasis, Konstantinos, et al. (2015) "Deep convolutional neural networks for efficient vision based tunnel inspection." 2015 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP). IEEE.

[19] He, Kaiming, et al. (2016) "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition.

[20] Moore, Mark, et al. (2001) “Reliability of visual inspection for highway bridges”, volume I. No. FHWA-RD-01-105. Turner-Fairbank Highway Research Center.