Summer 2020

Computer vision based deep learning models for cyber physical systems

Muhammad Monjurul Karim

COMPUTER VISION BASED DEEP LEARNING MODELS FOR CYBER PHYSICAL SYSTEMS

by

MUHAMMAD MONJURUL KARIM

A THESIS

Presented to the Graduate Faculty of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

in

SYSTEMS ENGINEERING

2020

Approved by:

Ruwen Qin, Advisor
Steven Corns
Cihan H. Dagli

PUBLICATION THESIS OPTION

This thesis consists of the following two articles, formatted in the style used by the Missouri University of Science and Technology:

Paper I, found on pages 5–24, has been published in the proceedings of 25th International Conference on Production Research, Chicago, USA, in August 2019.

Paper II, found on pages 25–43, has been published in the proceedings of 2019 Complex Adaptive Systems Conference, PA, USA, in November 2019.

ABSTRACT

Cyber-Physical Systems (CPSs) are complex systems that integrate physical systems with their counterpart cyber components to form a close loop solution. Due to the ability of deep learning in providing sensor data-based models for analyzing physical systems, it has received increased interest in the CPS community in recent years. However, developing vision data-based deep learning models for CPSs remains critical since the models heavily rely on intensive, tedious efforts of humans to annotate training data. Besides, most of the models have a high tradeoff between quality and computational cost. This research studies deep learning algorithms to achieve affordable and upgradable network architecture which will provide better performance. Two important applications of CPS are studied in this work. In the first study, a Mask Region-based Convolutional Neural Network (Mask R-CNN) was adopted to segment regions of interest from surveillance videos of manufacturing plants. Then, the Mask R-CNN model was modified to have consistent detection results from videos using temporal coherence information of detected objects. This method was extended to the second study, a task of bridge inspection to detect and segment critical structural components. A cellular automata-based pattern recognition algorithm was integrated with the Mask R-CNN model to find the crack propagation rate in the structural components. Decision-makers can make a maintenance decision based on the rate. A discrete event simulation model was also developed to validate the proposed methodology. The work of this research demonstrates approaches to developing and implementing vision data-based deep neural networks to make the CPS more affordable, scalable, and efficient.

ACKNOWLEDGMENTS

I would like to express the deepest appreciation to my advisor, Dr. Ruwen Qin, for her continuous support throughout my master’s study and research. She was always patient and expertly guided me. Her motivation and enthusiasm always kept me engaged in my research. Without her, I couldn’t have reached this stage.

I would also like to extend my sincere gratitude to Dr. Dagli and Dr. Corns. It was a great experience to work with them during all the course works. Their comments and discussion during classes were very helpful. Their mentoring and encouragement was especially valuable. I am also grateful to Dr. Zhaozheng Yin and Dr. Genda Chen for their support and insightful comments about my research.

Research work of this MS thesis is partially supported 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 and Grant No. 69A3551747107 through Mid-America Transportation Center (MATC). This work is also partially supported by Intelligent Systems Center (ISC) and the Department of Engineering Management and Systems Engineering at Missouri University of Science and Technology.

I am indebted to my parents, brothers, and sister for their unconditional love and support throughout my life. I would like to especially thank my younger brother, Rahad, for his continuous encouragement to do my research. And last but not the least, I would like to thank my beloved wife, Nishu, for believing in me, for being there for me during all the ups and downs throughout the journey.

TABLE OF CONTENTS

SECTION

LIST OF ILLUSTRATIONS

SECTION Page

PAPER I

PAPER II

LIST OF TABLES

PAPER I Page

PAPER II

1. INTRODUCTION

Cyber-physical systems (CPS) are complex engineering systems integrated with physical and cyber processes. One of the prospective methods of adding new capabilities to the cyber process of CPS is the use of computer vision because of their exquisite capability of integrating with physical systems [1]. Computer vision is the science that aims to mimic the capability of human eyes to see and sense the world. This science uses computational methods to extract the required information from visual images. Smart manufacturing, smart transportation, smart infrastructure systems are some of the applications that utilize video surveillance and inspection video data. For many decades researchers are actively researching these fields. Therefore, many applications of computer vision have been developed [2, 3, 4]. However, computer vision in developing complex adaptive cyber-physical systems has not been explored remarkably. Therefore, this field requires further exploration.

In this thesis, vision data-based deep learning models were developed and integrated with existing engineering systems to turn those systems into complex adaptive cyber-physical systems to tackle complex societal problems in two important applications. Figure 1 shows how our cyber models are interconnected with the physical process to control the physical world. This thesis chose to utilize the recent success of the deep learning method [5, 6, 7]. The recent success of deep learning methods brought new capabilities to develop new artificial intelligence, and machine learning techniques to tackle various socio-technical problems [8]. The availability of parallel computation coupled with the large-scale labeled datasets and the recent breakthrough of the deep learning algorithms reinforce the idea of developing vision data-based deep learning models to create complex adaptive cyber-physical systems.

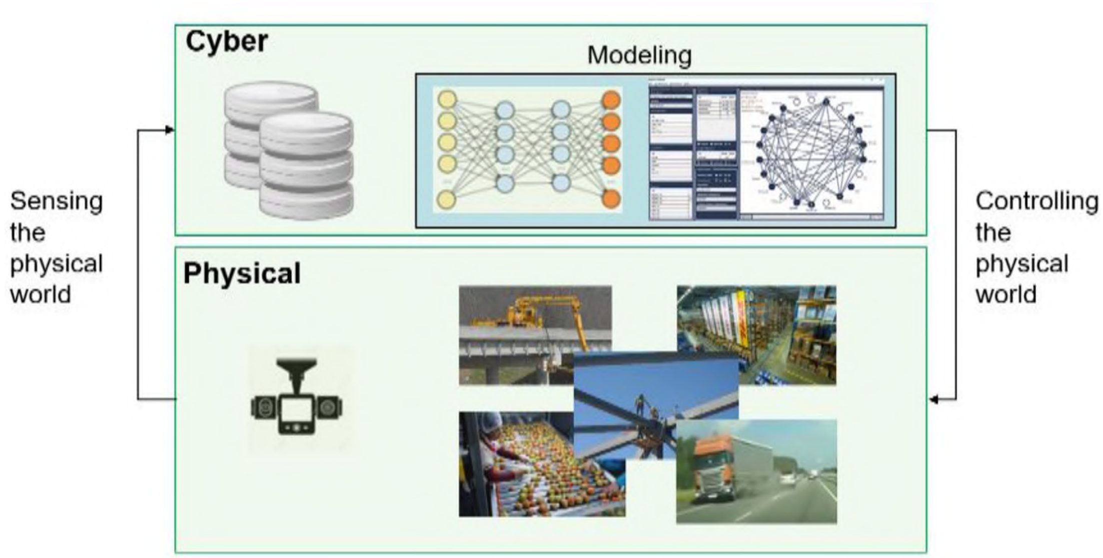


Figure 1. Cyber-Physical systems

Artificial intelligence assisted computer vision platforms are capable of functioning in increasingly complex environments. Artificial intelligence along with operators' skills leads to better efficiency and fewer errors. However, all machine vision platforms cannot provide similar benefits. Computer vision can yield the best result when strategically integrated into the intended operation to enhance both the machine and human intelligence simultaneously. Successful integration of computer vision, machine intelligence, and human intelligence can overcome the limitation of manual quality control, tracking inventories, and tracking industrial tools or objects in industrial applications. Therefore, in this thesis, a complex system was developed using a regional-based deep learning algorithm to detect and track industrial objects in complex manufacturing scenes. Mask Region-based Convolutional Neural Network (Mask R-CNN) [9] object detector was used to detect and segment important industrial objects from video frames. Temporal coherence analysis was added in the post-processing phase to further improve the segmentation quality. To train the model, transfer learning was used to make the model adaptable to new tasks. This also increases the affordability of the model as it requires only a few annotated training data.

The developed computer vision-based deep learning model is not only applicable to the manufacturing industry but also helps many other application fields. Inspecting complex engineering systems like bridges, buildings, pipelines, etc. are a few examples of such applications. Complex engineering infrastructure systems deteriorate over time, their proper inspection, monitoring, and maintenance are becoming very important. The traditional practice depends on the periodic visual inspection by humans, which is inadequate because inspection reports vary significantly among different inspectors due to their individually varying educational background, experiences, and physical conditions. Besides, inspection data collection is time-consuming, it requires dangerous field activities, sometimes need to block traffic. Therefore, to overcome these challenges this thesis develops a visual data based deep learning model that can analyze the data to detect critical structural components using the data collected by unmanned aerial vehicles (UAV). A self-organizing cellular automata-based pattern recognition algorithm was added with the deep neural model to find out the crack propagation rate in the structural components. A discrete event simulation model was also developed to validate the proposed inspection model.

The remaining part of this thesis is organized as below. Section 2 describes vision-based deep learning model in the manufacturing field. Section 3 describes the modeling and simulation of a robotic bridge inspection. Finally, Section 4 draws conclusions and suggests future research.

PAPER

I. A REGION-BASED DEEP LEARNING ALGORITHM FOR DETECTING AND TRACKING OBJECTS IN MANUFACTURING PLANTS

Muhammad Monjurul Karim, David Doell, Ravon Lingard, Zhaozheng Yin, Ming C. Leu, Ruwen Qin

Missouri University of Science and Technology, Rolla, MO 65409

ABSTRACT

In today's competitive production era, the ability to identify and track important objects in a near real-time manner is greatly desired among manufacturers who are moving towards the streamline production. Manually keeping track of every object in a complex manufacturing plant is infeasible; therefore, an automatic system of that functionality is greatly in need. This study was motivated to develop a Mask Region-based Convolutional Neural Network (Mask R-CNN) model to semantically segment objects and important zones in manufacturing plants. The Mask R-CNN was trained through transfer learning that used a neural network (NN) pre-trained with the MS-COCO dataset as the starting point and further fine-tuned that NN using a limited number of annotated images. Then the Mask R-CNN model was modified to have consistent detection results from videos, which was realized through the use of a two-staged detection threshold and the analysis of the temporal coherence information of detected objects. The function of object tracking was added to the system for identifying the

misplacement of objects. The effectiveness and efficiency of the proposed system were demonstrated by analyzing a sample of video footage.

1. INTRODUCTION

Visually finding an object in a complex manufacturing plant is a basic requirement of various industrial tasks like quality management, packaging, and sorting to name a few. Moreover, according to the industry 4.0 paradigm, monitoring objects and tracking their position in real time are needed for controlling production processes [1]. This capability also facilitates the recognition of human-object interaction, which will help to make machines and components become autonomous and self-organizing, thus reducing the manufacturing complexity [2, 3]. However, manually keeping track of objects lacks efficiency and reliability. Therefore, an automatic system of that functionality is greatly in need.

Automated object detection and tracking in complex manufacturing scenes are very important for developing a smart manufacturing industry, however, this remains a very challenging task. Researchers adopted some traditional solutions for this task including the use of weight or magnetic sensors [4, 5, 6]. The radio frequency identification (RFID) technology is also commonly used to track objects [7, 8, 9]. This technology requires an RFID active tag attached with the object, a tag reader, and radio communication between them, thus requiring a large initial cost. Meanwhile, to attach tags to every tools and objects in a factory is unrealistic. A more practical approach is computer vision that does not require attaching any material or sensors to objects of tracking. Computer vision detects and tracks objects through analyzing the video data, for example, using deep learning [10, 11]. Region-based convolutional neural network (R-CNN) has been shown to be effective in detecting and localizing objects in images [12]. Faster R-CNN proposed by He et al. [13] is a feature extractor that uses a region proposal network (RPN) to generate region proposals instead of using traditional selective search [14]. The RPN simultaneously regresses region bounding boxes and detection scores of an object. Mask R-CNN [15] is an extension of faster R-CNN, which performs the region segmentation at the pixel level. Recently, the YOLO [16] and SSD [17] use a single network which don't have the RPN and RoI-pooling layers, thus faster compared to Faster R-CNN and Mask R-CNN. However, YOLO and SSD are outperformed by Mask R-CNN and Faster R-CNN in detecting small objects.

Abovementioned deep learning algorithms work well in detecting objects in static images. Yet, results may not be consistent when they are applied to videos. Therefore, the temporal coherence of an object in successive frames has been introduced to address the issue of inconsistent detection [18, 19, 20], wherein the tubelet and optical flow are used to propagate features from one frame to another. However, the approaches in the literature are computationally expensive due to the requirement for repeated motion estimation and feature propagation, making the solution process very slow. Seq-NMS [21] has modification only in the post-processing phase and, thus, it is faster than the algorithms in [18, 19, 20]. Yet Seq-NMS tends to increase the number of false positive detections because it does not put a penalty on these detections or add additional constraints to prevent the occurrence.

This paper presents a study that extended Mask R-CNN by referring to the temporal coherence information of objects in videos and implementing a two-staged detection threshold. The temporal information includes high scoring objects in neighboring frames and their spatial locations. The two-staged detection threshold was introduced to boost up weak detections in a frame by referring to objects with high detection scores in neighboring frames. The spatial locations of these objects were used to prevent the propagation of false positive detections to other frames. This study further created the ability to track the location of any detected object and notify users if the object is not in the right place for it. In implementation of the proposed method, transfer learning [22] was used to adapt a deep learning feature extractor to the application setting.

The remainder of this paper is organized as follow: Section 2 delineates the proposed method for object detection and tracking, followed by examples illustrating the implementation of the method. Results from the examples are illustrated in Section 4. Conclusions and future work are summarized at the end, in Section 5

2. METHODOLOGY

The proposed framework for the object detection and tracking system is illustrated in Figure. 1. The system can use the plant's own Closed Circuit TV (CCTV) or surveillance cameras to capture videos of the work floor. Video streams of a monitored area are fed to the system. The classifier of the system uses a deep learning algorithm to semantically detect objects in that area. Then, the initial detection result is further refined by referring to the temporal coherence information of objects in videos. The system measures the distance between the location of each detected object and the location for the object in the designated zone. If the measured distance is larger than the pre-specified threshold value for the object, indicating that the object is outside the zone, a notification will be generated and sent to users through an interface. Provided with this system, users can track every object and find the location of it when the object is misplaced. The deep learning algorithm of object detection and tracking, which is the focus of this paper, is discussed in the following.

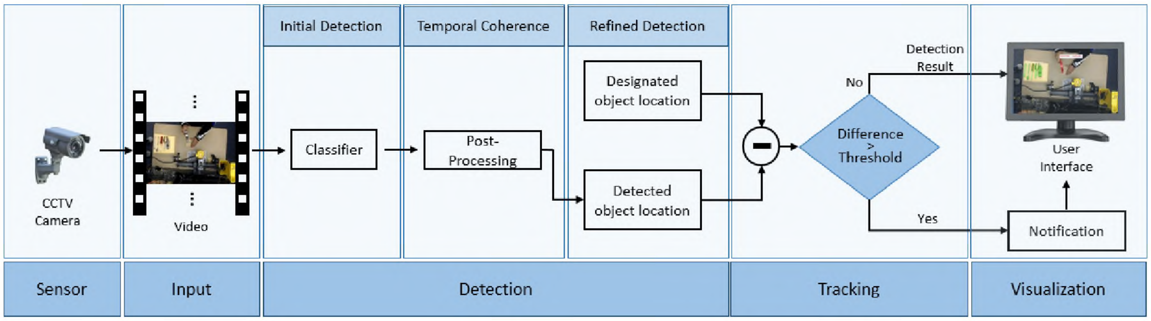


Figure 1. Schematic diagram of the object detection and tracking system

2.1. DEVELOPMENT OF A REGION BASED DEEP LEARNING ALGORITHM BASED ON TRANSFER LEARNING

Region-based CNN (R-CNN) has been shown to be effective in detecting and localizing objects in images. Mask R-CNN, a type of R-CNN performing the region segmentation at the pixel level, was chosen as the segmentation tool by this study. Figure 2 illustrates the structure of Mask R-CNN. Having an architecture of ResNet [23] based Feature Pyramid Network (FPN), the backbone of the network is a feature extractor that generates the feature map of each input image. A region proposal network (RPN) creates region of interests (ROIs) and extracts them from the feature map. The extracted feature maps are further aligned with the input image and converted into fixed size feature maps by a layer named Region of Interests Align (RoIAlign). The fixed-size feature maps of RoIs are fed into two independent branches: the network head branch performing classification and bounding box generation, and the mask branch for independently generating instance masks. Interested readers can refer to [15] for details.

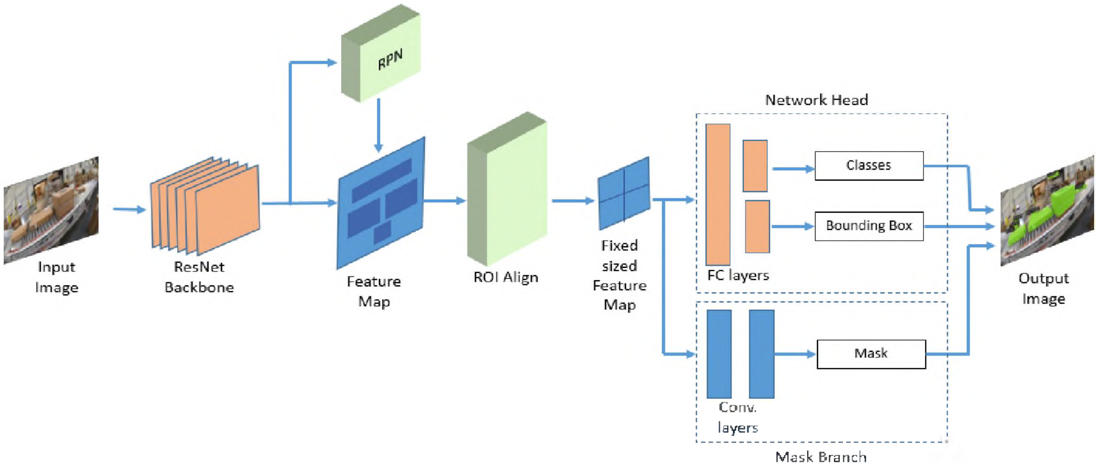


Figure 2. Architecture of Mask R-CNN

In this study, the Mask R-CNN was initialized by adopting the ResNet-50 feature extractor [23] whose weights have been pre-trained on the Microsoft COCO dataset [24] that has more than 120,000 labeled images and contains around 1.5 millions of object instances in 80 categories. Then, transfer learning was used to adapt the ResNet-50 feature extractor to the specific setting of this study. Specifically, the ResNet-50 was fine-tuned using a small set of training images collected from the intended manufacturing application. The ground truth of the training dataset was created by manually annotating the images with class labels. The training was a two-stage process. In the first stage, the network head and the mask branch were trained while all layers before the head were fixed. In the second stage, besides the network head and the mask branch the last few layers of the ResNet Backbone (C5) were trained as well.

2.2. TEMPORAL COHERENCE WITH A TWO-STAGED DETECTION THRESHOLD

False detections can be reduced by incorporating the temporal coherence information of objects in successive frames. The temporal information used by this study include objects with high detection scores in preceding frames and their spatial locations. The temporal coherence of objects in videos was incorporated in the post-processing phase of the Mask R-CNN.

Consider a single video clip that consists of N frames, index by i. In each frame the detector returns M_i objects indexed by j. An object in a frame is highly likely present in the neighboring frames within a range of displacement with similar confidence. Under this assumption, a two-staged detection threshold was introduced in this study to propagate detection results from one frame to succeeding frames. Let $o_{i,j}$ designate object j in frame i. The center of the bounding box for $o_{i,j}$ is specified by its coordinates $C_{i,j} \coloneqq (x_{i,j}, y_{i,j})$. In p frames, $C_{i,j}$ may shift to a surrounding pixel with a spatial displacement of $(p\Delta x, p\Delta y)$ where Δx and Δy are the average displacement on x and y axes, respectively.

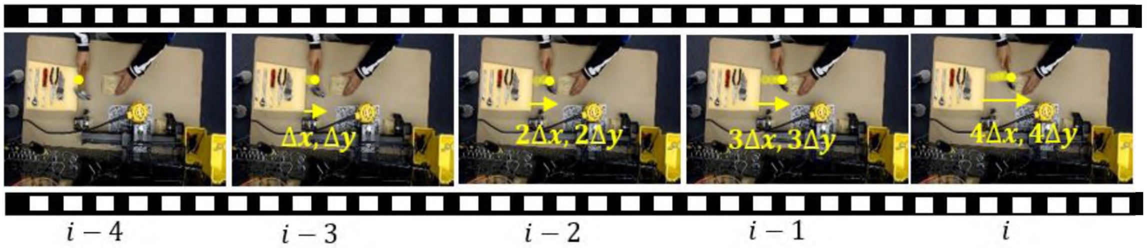
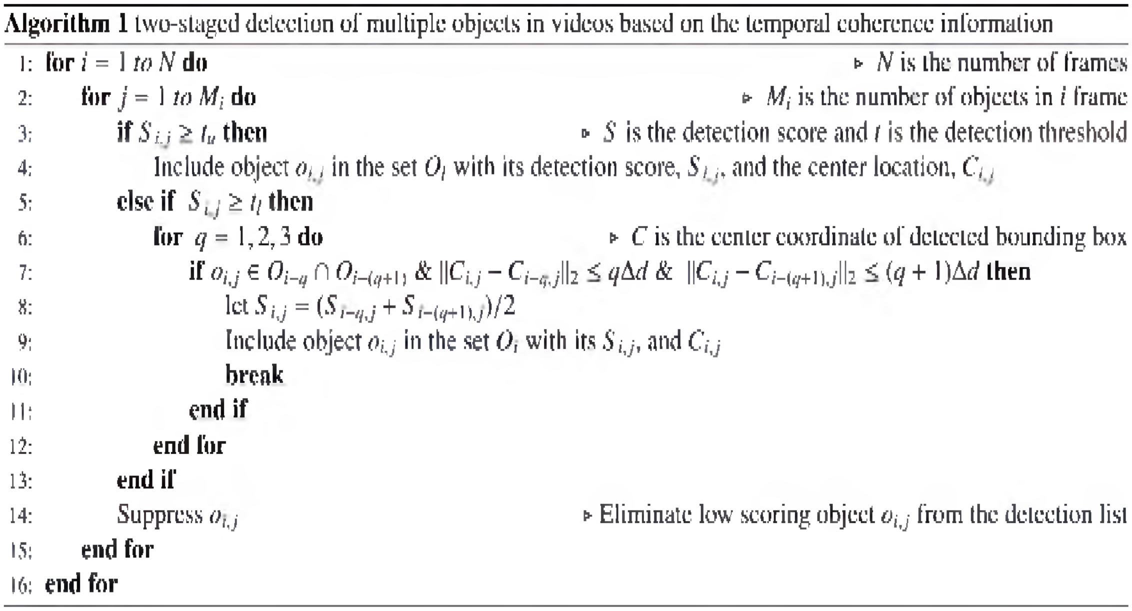


Figure 3. An illustration of spatial displacements. The yellow dot represents the center location of the bounding box for an object. In each frame it gets displaced by $(\Delta x, \Delta y)$, approximately

The detection score for object j in frame i is $S_{i,j}$. The detection threshold is a range $[t_l, t_u]$. The detector immediately returns a positive detection if $S_{i,j} > t_u$. Let O_i be the set of such detected objects. The detection score and the center location of these objects in frame i, $(S_{i,j}, C_{i,j})$, are stored for analyzing the succeeding four frames. If $t_l \leq S_{i,j} \leq t_u$, the existence of this weakly detected object $o_{i,j}$ is checked by referring to a pair of preceding successive frames up to three times, starting from the nearest pair (frames $i-1$ and $i-2$) to the farthest pair (frames i−3 and $i-4$). That is, if $o_{i,j}$ is found in both O_{i-1} and O_{i-2}, and the spatial displacements of $C_{i,j}$ from frame $i-1$ and $i-2$ are within $(\Delta x, \Delta y)$ and $(2\Delta x, 2\Delta y)$, respectively, this weakly detected object is added to O_i and the detection score of it is updated by taking the average of $S_{i-1,j}$ and $S_{i-2,j}$. Otherwise, $o_{i,j}$ is searched in O_{i-2} and O_{i-3} and the displacements of $C_{i,j}$ from frames $i-2$ and $i-3$ are measured to determine if it is a positive detection. $o_{i,j}$ will be searched from O_{i-3} and O_{i-4} if needed. If $o_{i,j}$ is not found to be a positive detection after three times of time coherence analysis, it is not reported as a positive detection. It is noticed that searching an object in pairs of successive frames will minimize the risk of propagating false positive detection to succeeding frames. The algorithm for the two-staged process for detecting multiple objects from videos bases on the temporal coherence information of objects is summarized as the pseudocode below:



```
Algorithm 1 two-staged detection of multiple objects in videos based on the temporal coherence information
for $i = 1$ to $N$ do                                                ▷ $N$ is the number of frames
    for $j = 1$ to $M_i$ do                                          ▷ $M_i$ is the number of objects in $i$ frame
        if $S_{i,j} \geq t_u$ then                 ▷ $S$ is the detection score and $t$ is the detection threshold
            Include object $o_{i,j}$ in the set $O_i$ with its detection score, $S_{i,j}$, and the center location, $C_{i,j}$
        else if $S_{i,j} \geq t_l$ then
            for $q = 1, 2, 3$ do                           ▷ $C$ is the center coordinate of detected bounding box
                if $o_{i,j} \in O_{i-q} \cap O_{i-(q+1)}$ & $\|C_{i,j} - C_{i-q,j}\|_2 \leq q\Delta d$ & $\|C_{i,j} - C_{i-(q+1),j}\|_2 \leq (q+1)\Delta d$ then
                    let $S_{i,j} = (S_{i-q,j} + S_{i-(q+1),j})/2$
                    Include object $o_{i,j}$ in the set $O_i$ with its $S_{i,j}$, and $C_{i,j}$
                    break
                end if
            end for
        end if
        Suppress $o_{i,j}$                         ▷ Eliminate low scoring object $o_{i,j}$ from the detection list
    end for
end for
```

Figure 4. Pseudocode showing the two-staged detection method of multiple objects in videos based on the temporal coherence information

2.3. OBJECT TRACKING

The bounding box for the designated region for object j is denoted as $R_j \coloneqq \{w_j, h_j, (x_j^R, y_j^R)\}$, where w_j and h_j are the width and height of the box, respectively, and (x_j^R, y_j^R) is the coordinates of the center. If the center of the bounding box for object j in frame i,$C_{i,j}$, is outside R_j, A notification label $P_{i,j}$ is generated:

$$P_{i,j} = \begin{cases} 1 & \text{if } |x_{i,j} - x_j^R| > 0.5w_j \text{ or } |y_{i,j} - y_j^R| > 0.5h_j \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

If at least one notification label returns 1 (i.e., there exists $P_{i,j} = 1$ for $j \in \{1,2,\ldots,M_i\}$), a warning is sent to the user through the interface.

3. IMPLEMENTATION DETAILS

The proposed method was evaluated through experiments in two manufacturing scenes (a workstation and a production line) and under three different lighting scenarios: normal, underexposed and overexposed lighting conditions. The illumination level at different locations of manufacturing plants such as warehouse, work area, assembly area, and inspection area, can be very different. Therefore, this study tested the impact of lighting condition to detection results.

3.1. EXPERIMENTS AND DATA COLLECTION

A workstation was replicated in the lab and a camera was installed on the top of the workstation. The camera captured videos of workers when they operate at that workstation with a frame rate of 30fps. The videos are then converted to images to create the training dataset. A dataset (A) of 4,405 frames was acquired in this lab setup. Then, this dataset was duplicated to create a new dataset (B) by reducing the brightness of the images by 90% . Similarly, another dataset (C) was created by increasing the brightness level by 70%.

Using a video stream of an actual production line available on YouTube, three more datasets (D, E, and F) were collected. Dataset D is the original video that consists of 400 frames under a normal lighting condition. Datasets E and F were created in the same manner as datasets B and C, respectively. The size of all the image data is 1024 x 578 pixels and the resolution is 96 dpi.

3.2. MODEL TRAINING AND FINE-TUNING

The ResNet-50 feature extractor was initialized with weights pre-trained on the Microsoft COCO dataset. The model was fine-tuned using a small set of training dataset composed of 40 images from the workstation dataset. The ground truth data of the training dataset were carefully created by manually annotating the images with 6 class labels, namely hammer, screwdriver, wrench, ratchet, plier, and allen key. In the first stage, the network head and the mask head was trained for 30 epochs and all the parameters in the previous layers were fixed. In the second stage, in addition with the heads, the ResNet Backbone C5 were trained for 30 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 30 epochs of training. The batch size of one image was used on a single NVIDIA Geforce GTX 1080 Ti GPU for this fine-tuning process that took about 14 hours to complete.

The model was further fine-tuned for the production line using a training dataset of 10 images. This dataset has only one class label, package. But the production line had more complex background than the workstation. This time, the network head and the mask head were trained for 25 epochs and all other layers were the same as those obtained from the first stage training for the workstation example.

4. RESULT AND DISCUSSION

The object tracking system was evaluated on a workstation with the following configuration: a 2.90 GHz Intel Xeon W-2102 CPU with 4 CPU cores, 16GB of RAM and an NVIDIA Geforce GTX 1080 Ti GPU. In the evaluation, the lower boundary of detection threshold, tl, was 0.5 and the upper boundary, tu was 0.8. To quantify the model effectiveness, a validation dataset of 280 images was created by taking images from the normal, overexposed, and underexposed lighting conditions. 240 out of 280 images were relevant to the workstation scene and the remaining 40 images were from the production line scene. In total 1691 ground truth labels were considered in the evaluation.

4.1. QUANTITATIVE RESULT

Intersection over Union (IoU) is the intersection between the predicted bounding box and the ground truth bounding box over the union of them. This ratio was used to determine whether a predicted object can be considered as a correct detection. In the experimental studies of this paper, the IoU value must exceed 0.60 to be considered as a correct detection.

Table 1 compares the object detection ability of Mask R-CNN without temporal coherence to the one with temporal coherence under each of the three lightness conditions. Three classic assessment matrices have been used. The assessment metrics are in this comparison:

- Precision: it counts the number of correctly predicted classes out of the total number of predictions

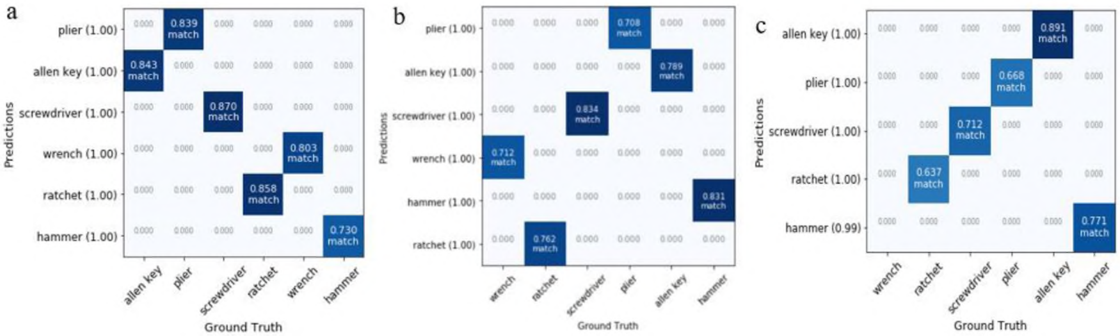
- Recall: it counts the number of correctly predicted classes out of total number of ground-truth objects
- F1-Score: it is the harmonic mean of precision and recall

Table 1. Results on Mask R-CNN model and Mask R-CNN + temporal coherence

	Illumination level	Precision	Recall	F1
Mask R-CNN	Normal	0.963	0.950	0.957
	Underexposed	0.935	0.922	0.928
	Overexposed	0.733	0.493	0.590
Mask R-CNN + Temporal coherence	Normal	0.991	0.979	0.985
	Underexposed	0.993	0.929	0.960
	Overexposed	0.727	0.500	0.593

From Table 1 it can be seen that the Mask R-CNN model obtained a high precision (96.3%), recall (95%), and F1- Score (95.7%) under the normal lighting condition. These three scores dropped by around 3% under the underexposed lighting condition, and over 20% under the overexposed condition. Adding the temporal coherence information to the Mask R-CNN increased the precision for about 3% under the normal lighting condition and 6% under the underexposed condition. The improvements are due to the fact that the temporal coherence information was used for lowering the amount of false positive detections. The improvement of recall was near 3% under the normal lighting condition and only 0.5% under the underexposed condition, indicating the temporal coherence information helps improve the ability to correctly detect more relevant objects under normal lighting condition. However, the addition of temporal coherence to the Mask R-CNN did not improve either the precision or recall under the overexposed lighting condition. This is because edges of objects may not be differentiable from their background under an overexposed condition, as shown in the left column of Figure 7(b). The detection result may get worse after applying temporal coherence to the object detection if there were false detections in several continuous frames.

To evaluate the pixel-wise accuracy of the proposed algorithm, an overlapping grid of ground truth objects and corresponding predictions was calculated under various lighting conditions.



Predictions \ Ground Truth	allen key	plier	screwdriver	ratchet	wrench	hammer
plier (1.00)	0.000	0.839 match	0.000	0.000	0.000	0.000
allen key (1.00)	0.843 match	0.000	0.000	0.000	0.000	0.000
screwdriver (1.00)	0.000	0.000	0.870 match	0.000	0.000	0.000
wrench (1.00)	0.000	0.000	0.000	0.000	0.803 match	0.000
ratchet (1.00)	0.000	0.000	0.000	0.858 match	0.000	0.000
hammer (1.00)	0.000	0.000	0.000	0.000	0.000	0.730 match

Predictions \ Ground Truth	wrench	ratchet	screwdriver	plier	allen key	hammer
plier (1.00)	0.000	0.000	0.000	0.708 match	0.000	0.000
allen key (1.00)	0.000	0.000	0.000	0.000	0.789 match	0.000
screwdriver (1.00)	0.000	0.000	0.834 match	0.000	0.000	0.000
wrench (1.00)	0.712 match	0.000	0.000	0.000	0.000	0.000
hammer (1.00)	0.000	0.000	0.000	0.000	0.000	0.831 match
ratchet (1.00)	0.000	0.762 match	0.000	0.000	0.000	0.000

Predictions \ Ground Truth	wrench	ratchet	screwdriver	plier	allen key	hammer
allen key (1.00)	0.000	0.000	0.000	0.000	0.891 match	0.000
plier (1.00)	0.000	0.000	0.000	0.668 match	0.000	0.000
screwdriver (1.00)	0.000	0.000	0.712 match	0.000	0.000	0.000
ratchet (1.00)	0.000	0.637 match	0.000	0.000	0.000	0.000
hammer (0.99)	0.000	0.000	0.000	0.000	0.000	0.771 match

Figure 5. Overlapping grid between ground truth and prediction under (a) normal, (b) underexposed, and (c) overexposed lighting conditions

Figure 5 shows three examples of the overlapping grid with each of them under one of the three lighting conditions. In Figure 5 the ground truth classes are listed on the horizontal axis, and on the vertical axis the predicted classes are listed in the decreasing order of detection probability. Each grid describes the IoU value of the detected class. It can be seen from Figure 5(a) and 5(b) that the IoU values for all detected classes are higher than 60% under the normal and underexposed lighting conditions. However, under the overexposed lighting condition, the class wrench is not listed in the vertical axis as its

IoU value is lower than 60%. Moreover, the IoU value of 4 classes (wrench, plier, screwdriver, and ratchet) out of 6 classes in this condition is lower than the corresponding values under the other two lighting conditions. The result shows that adding temporal coherence to the Mask R-CNN performs well in the pixel level segmentation under both normal and underexposed lighting conditions, but not under the overexposed lighting condition.

4.2. QUALITATIVE RESULT

Figure 6 illustrates how notifications were generated when objects were moved out of their designated area. The system highlighted an object with a red mask when it was moved to the outside of the designated region for it.

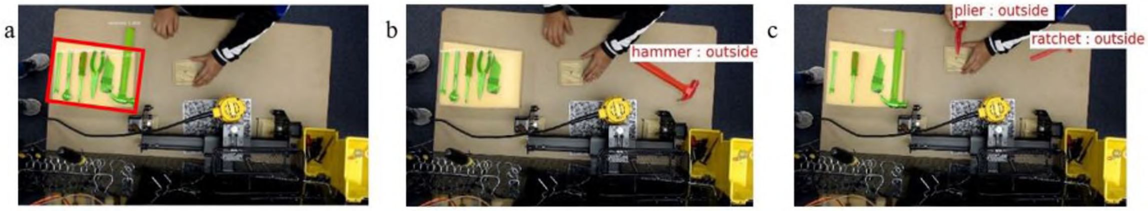


Figure 6. Setup of the workstation. (a) Six tools are highlighted with green mask. From left to right, the tools are wrench, ratchet, screwdriver, plier, Allen key and hammer. The red box indicates the designated area for these tools; (b) a notification is generated by highlighting the hammer with red mask as it gets out of its predefined designated area; (c) notifications generated for plier and ratchet

Figure 7(a), (b), and (c) show some successful examples of object detection by the proposed detecting and tracking system in the workstation scene under various lighting conditions. All these experiments were done in the workstation replicated in the lab. Under all three lighting conditions, the proposed system segmented objects successfully.

Masks tightly overlapped with the corresponding objects. No obvious false positives were found in those examples. Examples in Figure 7(d) show that there were no false positive detections in the production line scene. This is because the temporal information was used to suppress their appearance. The system also successfully generated notifications when objects were outside of the designated region.

Figure 8 illustrates some failed examples. Figure 8(a) and (b) are examples of false negative detections where some objects were not detected. It is found that false negative detections occurred when there was a motion blur in multiple successive frames, larger than the defined temporal window size.

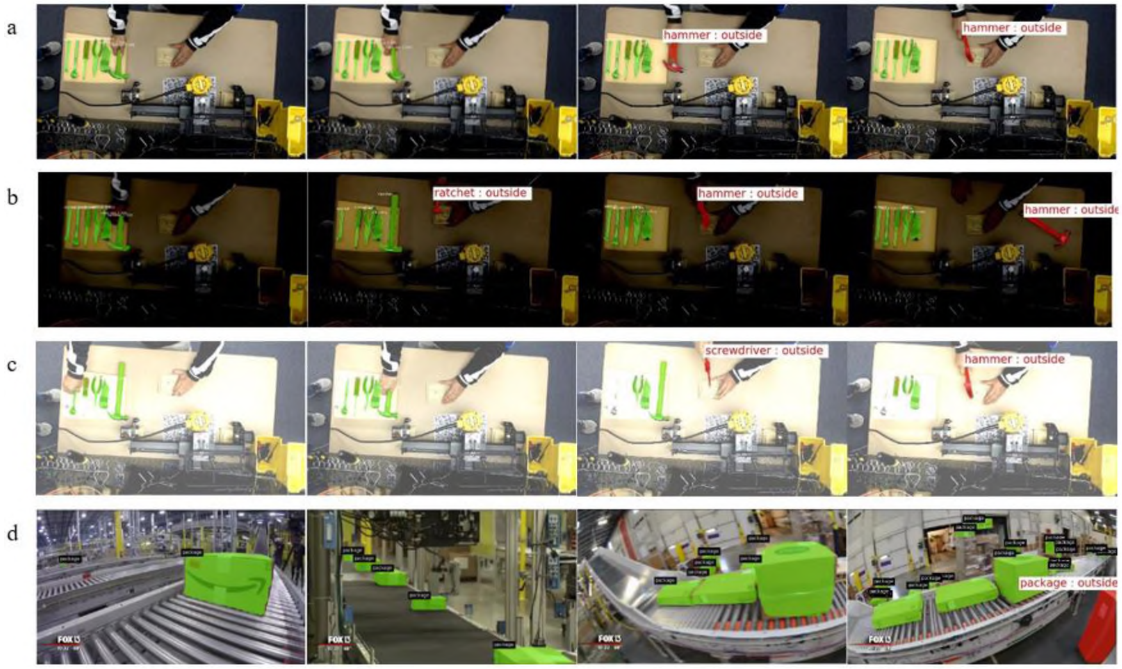


Figure 7. Successful examples of object detection using the proposed system. (a) and (d) are under the normal lighting condition; (b) is under underexposed lighting condition; and (c) is under overexposed lighting condition

Figure 8(a) shows such situations where the hammer (in the red bounding box) was not get detected because of motion blur. Moreover, the change of lighting condition may make objects unrecognizable, resulting detection failures. For example in Figure 8(b), in the overexposed lighting condition edges of wrench and ratchet were not recognizable, and under the underexposed lighting condition the plier handle was not recognizable. As a result the detector cannot detect these objects. Figure 8(c) illustrates another two examples of false positive detections where a screwdriver and a hammer were misclassified as a wrench and a plier, respectively. This is because a single camera cannot reveal the full appearance details of objects.

5. CONCLUSION

This paper presents a vision sensor based system for simultaneously detecting and segmenting industrial objects. This ability enables manufacturers to know the exact location of an object. The essence component of this system is an improved Mask R-CNN developed in the study of this paper. The post-processing phase of this network was modified to further refine the initial detection result using a two-staged detection threshold and the temporal coherence information of objects in successive frames. The temporal coherence method successfully recovers false negative detection to improve the detection result. The final detection result of an object was compared with its predefined location to know if a misplacement of the object from its original location was identified.

Results of the proposed algorithm are very promising to be used in real manufacturing settings. This algorithm achieved over 96% F1-score in normal and underexposed lighting conditions. Yet, detection quality needs to be improved under some challenging conditions such as: when motion blur is presented for a relatively long period of time; when the illumination level is too high; and when the camera viewpoint is limited. Future work would be focused on those matters to further refine the detection quality.

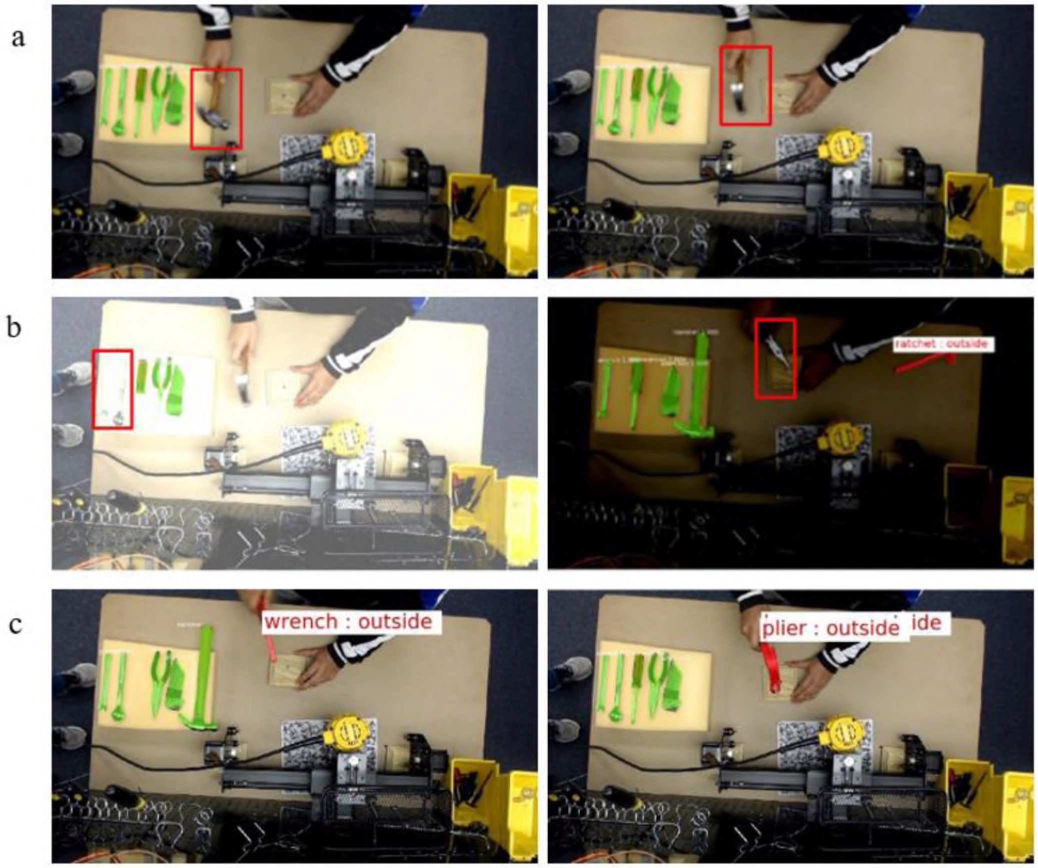


Figure 8. Overlapping grid between ground truth and prediction under (a) normal, (b) underexposed, and (c) overexposed lighting conditions

REFERENCES

[1] F. Almada-Lobo, “The Industry 4.0 revolution and the future of manufacturing execution systems (MES),” Journal of innovation management, 3(4)16-21, 2016.

[2] D. Gorecky, M. Schmitt, M. Loskyll, and D. Zühlke, “Human-machine-interaction in the industry 4.0 era,” In 2014 12th IEEE International Conference on Industrial Informatics (INDIN), pp. 289-294, 2014.

[3] J. Cannan and H. Hu, “Human-machine interaction (hmi): A survey,” University of Essex, 2011.

[4] W.C. Maloney, “Object tracking method and system with object identification and verification,” U.S. Patent No. 6,707,381, 2004.

[5] T. Paulsen, H. Meyer, and F. Arman, “System for tracking object locations using self-tracking tags,” U.S. Patent No. 7,119,687, 2006.

[6] S.B. Tantry, R. U. Mashruwala, B. A. Lozier, and R. L. Hess, “Object-oriented architecture for factory floor management,” U.S. Patent No. 5,398,336, 1995.

[7] K. Ding, P. Jiang, P. Sun and C. Wang, “RFID-enabled physical object tracking in process flow based on an enhanced graphical deduction modeling method,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, 47(11), 3006-3018, 2017.

[8] M. Liukkonen, “RFID technology in manufacturing and supply chain,” International Journal of Computer Integrated Manufacturing, 28(8) 861-880, 2015.

[9] J. Brusey and D. C. McFarlane, “Effective RFID-based object tracking for manufacturing,” International Journal of Computer Integrated Manufacturing, 22(7) 638-647, 2009.

[10] N. Cohen, O. Sharir, and A. Shashua, “On the expressive power of deep learning: A tensor analysis,” In Conference on Learning Theory, pp. 698-728, 2016.

[11] J. Wang, Y. Ma, L. Zhang, R. X. Gao, and D. Wu, “Deep learning for smart manufacturing: Methods and applications,” Journal of Manufacturing Systems, 48 144-156, 2018.

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 580-587, 2014.

[13] S. Ren, K. He, R. Girshick and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” In Advances in neural information processing systems, pp. 91-99, 2015.

[14] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, “Selective search for object recognition,” International journal of computer vision, 104(2) 154-171, 2013.

[15] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn,” In Computer Vision (ICCV), In proceedings of the IEEE International Conference on computer vision, pp. 2980-2988, 2017.

[16] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, “You only look once: Unified, real-time object detection,” In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 779-788, 2016.

[17] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, “Ssd: Single shot multibox detector,” In European conference on computer vision, pp. 21-37, Springer, Cham, 2016.

[18] K. Kang *et al*. “T-cnn: Tubelets with convolutional neural networks for object detection from videos,” IEEE Transactions on Circuits and Systems for Video Technology, 28(10) 2896-2907, 2018.

[19] X. Zhu, Y. Xiong, J. Dai, L. Yuan and Y. Wei, “Deep feature flow for video recognition,” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2349-2358, 2017.

[20] X. Zhu, Y. Wang, J. Dai, L. Yuan and Y. Wei, “Flow-guided feature aggregation for video object detection,” In Proceedings of the IEEE International Conference on Computer Vision, pp. 408-417, 2017.

[21] W. Han, P. Khorrami, T. L. Paine, P. Ramachandran, M. Babaeizadeh, H. Shi, and T. S. Huang, “Seq-nms for video object detection.” arXiv preprint arXiv:1602.08465, 2016.

[22] I. Goodfellow, Y. Bengio, and A. Courville, “Deep learning,” MIT press, 2016.

[23] K. He, X. Zhang, S. Ren and J. Sun, “Deep residual learning for image recognition,” In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778, 2016.

[24] T. Y. Lin *et al*., “Microsoft coco: Common objects in context, In European conference on computer vision, pp. 740-755, Springer, Cham, 2014.

II. MODELING AND SIMULATION OF A ROBOTIC BRIDGE INSPECTION SYSTEM

Muhammad Monjurul Karim, Cihan H. Dagli, Ruwen Qin

Department of Engineering Management and Systems Engineering, Missouri University of Science and Technology, Rolla, MO 65409

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.

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 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 infrastructure 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. [11] 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 [12, 13, 14]. 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. [15] monitored crack changes in the concrete structure. They focused on quantifying the periodic change in the cracks from multi-temporal images. Ito et al. [16] 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. [17] developed a two-staged threshold based object detection method that can detect multiple objects in an image. They used Mask R-CNN [18] 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.

In this paper a system is developed to eliminate the gap. 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 Figure 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.

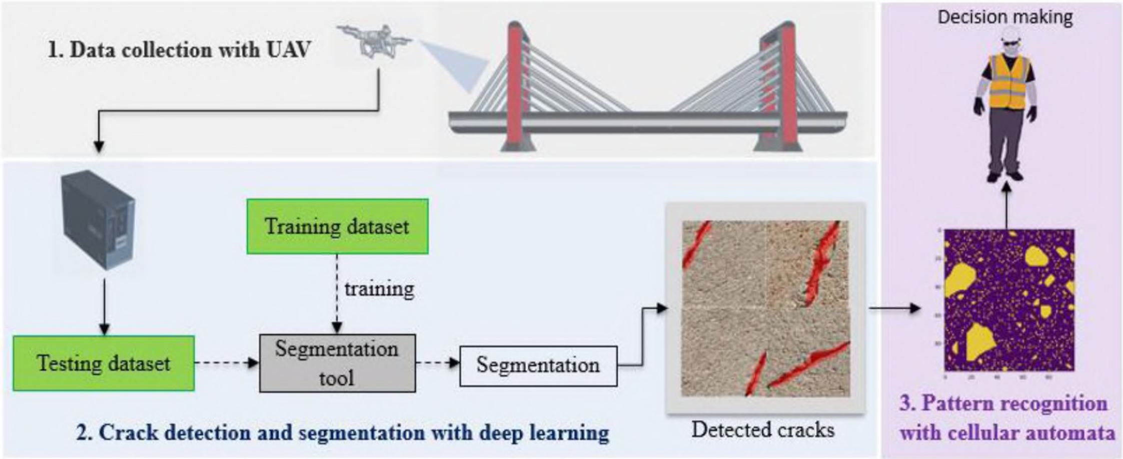

1. Data collection with UAV
Decision making
Training dataset
training
Testing dataset
Segmentation tool
Segmentation
2. Crack detection and segmentation with deep learning
Detected cracks
3. Pattern recognition with cellular automata

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

2.1. DEEP LEARNING ALGORITHM

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. Figure 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 [18] 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 $\emptyset$ that depends on the value of sites in some neighborhood around it. Figure 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.

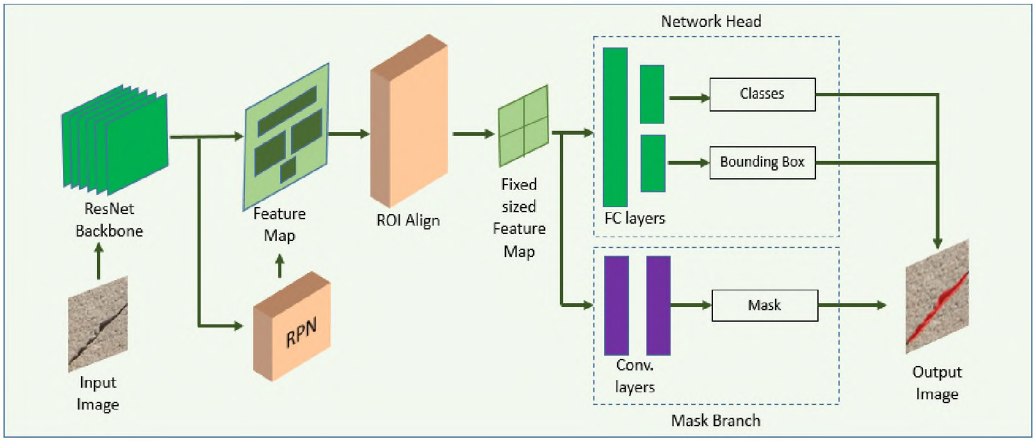
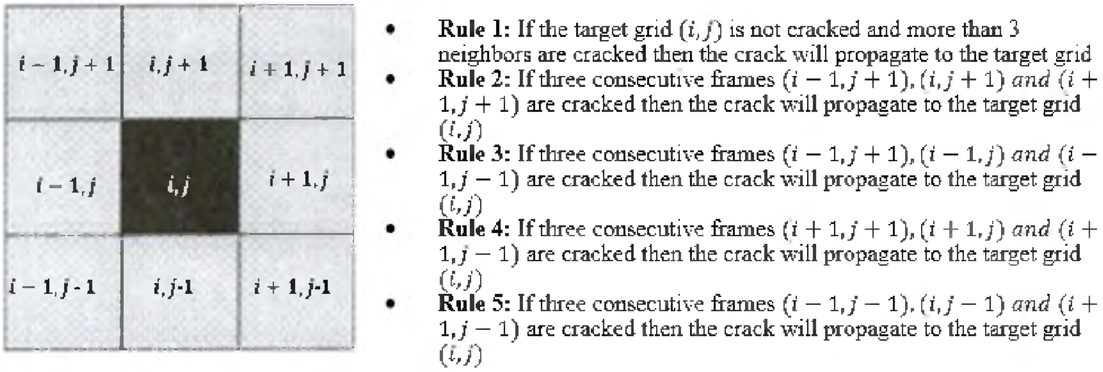


Figure 2. Architecture of the Mask R-CNN

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.



- **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)

Figure 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

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.

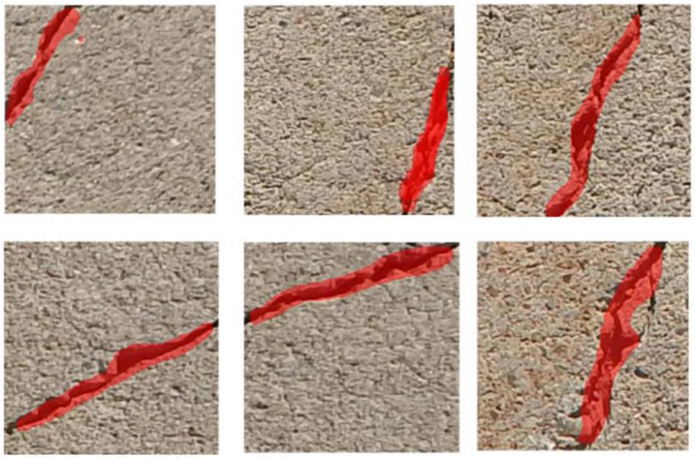


Figure 4. Examples of detected cracks with the deep learning algorithm

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. Figure 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 Figure 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.

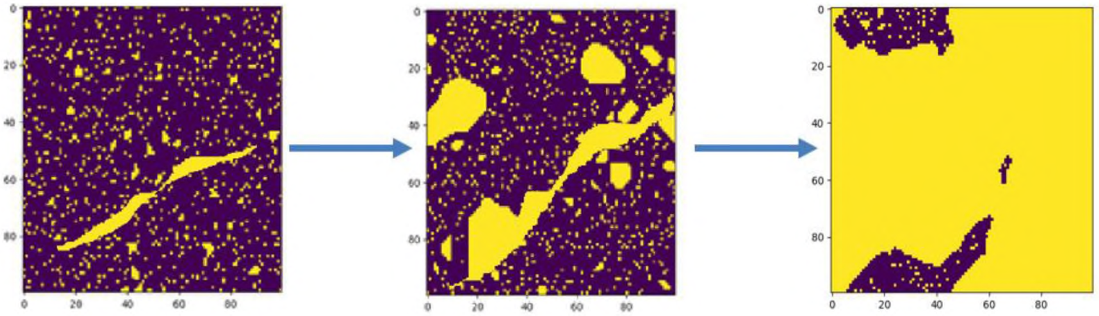


Figure 5. Crack propagation at 3 time steps. The yellow cells represent cracked cells. At each update interval, it gets propagated

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 UAV, deep learning machine and maintenance team are. For simulation individual logic has been developed for each agent.

4.1. LOGIC FOR UAV

Two different process flow diagram has been developed for inspecting two different bridge. Figure 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.

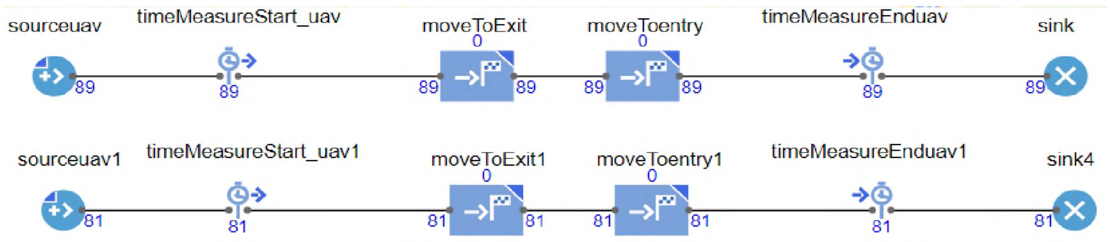


Figure 6. Logic for UAV operation

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.

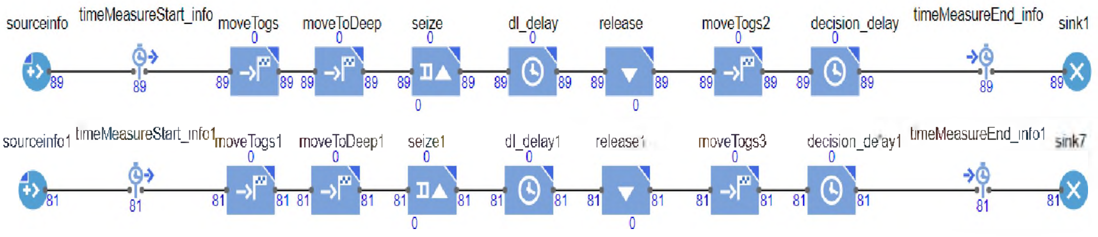


Figure 7. Logic for deep learning and decision making

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.

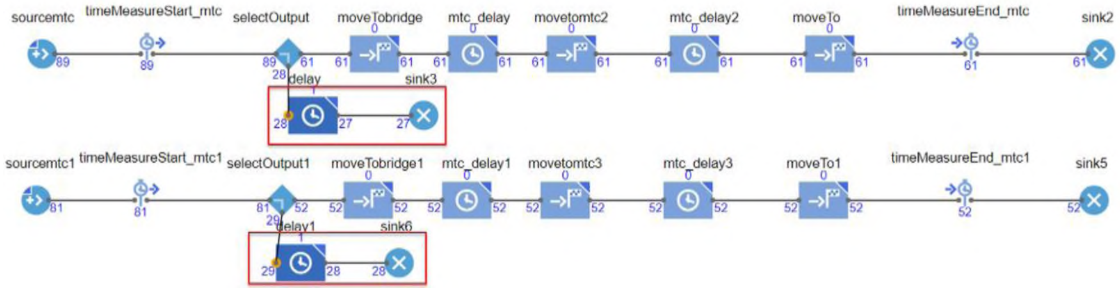


Figure 8. Logic for maintenance

4.4. STATISTICS REPORT

After simulating the model for five years, a statistical report has been generated in the AnyLogic. Figure 9 (a) represents the cycle time required for UAV inspection. UAV1 represents the inspection for bridge 1 and Uav2 represents inspection for bridge 2. Figure 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.

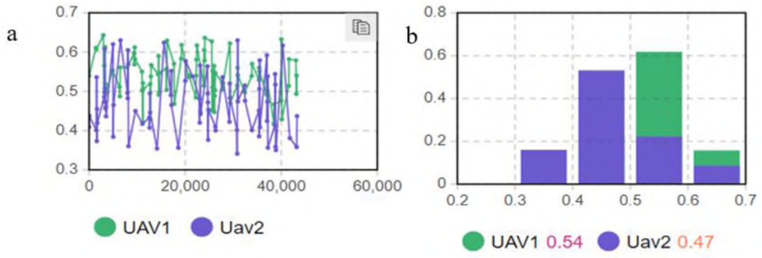


Figure 9. (a) cycle time of inspection (b) histogram of cycle times

Figure 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. Figure 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.

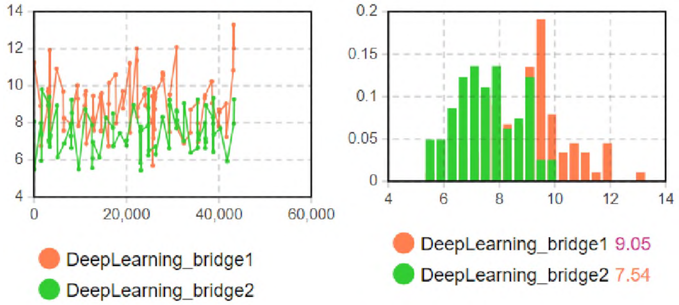


Figure 10. (a) Cycle time for deep learning and decision making (b) histogram of cycle times

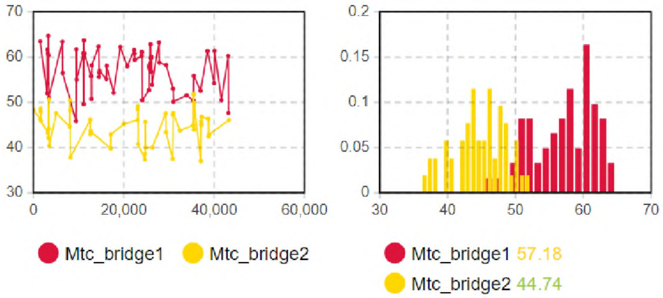


Figure 11. (a) Cycle time for maintenance (b) histogram of cycle times

Figure 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. Figure 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.

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.

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

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

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

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

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

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

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

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

[11] R. S. Lim *et al.*, "Developing a crack inspection robot for bridge maintenance." 2011 IEEE International Conference on Robotics and Automation. IEEE, 2011.

[12] S. N. Yu, J. H. Jang, and C. S. Han, "Auto inspection system using a mobile robot for detecting concrete cracks in a tunnel." Automation in Construction 16(3): 255-261, 2007.

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

[14] P. C. Tung, Y. R. Hwang, and M. C. Wu, "The development of a mobile manipulator imaging system for bridge crack inspection." Automation in construction 11 (6): 717-729, 2002.

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

[16] A. Ito, Y. Aoki, and S. Hashimoto, "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, 2002.

[17] M. M. Karim *et al.* "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, 2019.

[18] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," In Computer Vision (ICCV), In proceedings of the IEEE International Conference on computer vision, pp. 2980-2988, 2017.

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

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

SECTION

2. CONCLUSIONS AND FUTURE WORK

Computer vision-based deep learning models have been developed and integrated with complex engineering systems to develop complex adaptive cyber-physical systems. The developed systems have the capability of simultaneously detecting and segmenting objects in a complex scenario. The developed model showed promising success in the manufacturing industry. The system developed in this thesis will enable manufacturers to know the exact location of an object, which is almost impossible to track manually in a large manufacturing plant. Several new techniques were pioneered in the research. Temporal coherence analysis (TCA) was one of them. TCA was proposed to improve the efficiency of the developed system. Various lighting conditions were simulated to replicate the actual manufacturing settings and tested with the developed system. Promising results, including experimental results and theoretical analysis, demonstrate the prospect of using the proposed CPS in real manufacturing settings.

This research was extended to the problem domain of bridge inspection. A vision sensor based system was proposed to monitor and inspect bridges to detect and locate cracks in the bridges. A cellular automata-based pattern recognition algorithm was integrated to add the additional capability of determining the crack propagation rate from the visual images. The emergence behavior of the self-organizing cellular automata makes the system a complex adaptive system. To justify the effectiveness of the proposed system, a discrete event simulation model was also developed. The analysis with the simulation model shows that the proposed system can reduce 80% of the inspection time. This simulation model can also be used as a decision support tool for advanced analysis of the bridge inspection image data.

As vision data-based CPS is an emerging research topic, there are still many opportunities to conduct further research. Our future research work will focus on the following:

- The selection of neural network architecture for specific CPS still lacks a knowledge-driven method. This area of NN architecture selection requires further exploration. Our future work will focus on finding the best network architecture for CPS.
- Another important research direction is investigating the best system integration method in a CPS. Although deep NN provides advanced capabilities to the CPS, inappropriate system integration will underm the effectiveness of the deep learning in CPS. Therefore, there should be appropriate engineering methods and practices to effectively integrate deep learning methods into CPS.
- Another future direction of research is developing interactive deep learning models that keep humans in the loop. Traditional deep learning models are not adaptable to new data because it works like a black box on new data [10]. This does not take domain knowledge from the experts. Our future work will focus to develop deep NN that will take inputs from human experts during execution to update the model.

APPENDIX

COPYRIGHT INFORMATION

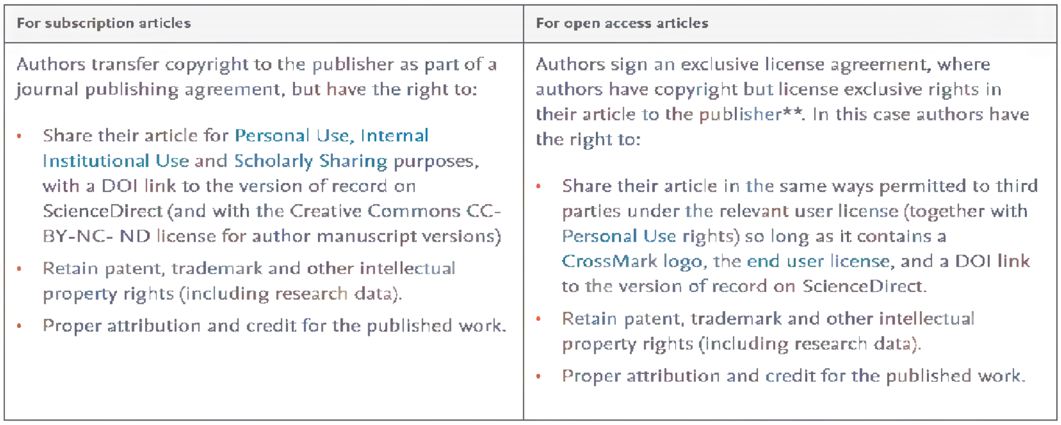
Copyright policy of Elseviar publisher is given below:

COPYRIGHT

Describes the rights related to the publication and distribution of research. It governs how authors (as well as their employers or funders), publishers and the wider general public can use, publish and distribute articles or books.

JOURNAL AUTHOR RIGHTS

In order for Elsevier to publish and disseminate research articles, we need publishing rights. This is determined by a publishing agreement between the author and Elsevier.



For subscription articles	For open access articles
Authors transfer copyright to the publisher as part of a journal publishing agreement, but have the right to: • Share their article for Personal Use, Internal Institutional Use and Scholarly Sharing purposes, with a DOI link to the version of record on ScienceDirect (and with the Creative Commons CC-BY-NC- ND license for author manuscript versions) • Retain patent, trademark and other intellectual property rights (including research data). • Proper attribution and credit for the published work.	Authors sign an exclusive license agreement, where authors have copyright but license exclusive rights in their article to the publisher**. In this case authors have the right to: • Share their article in the same ways permitted to third parties under the relevant user license (together with Personal Use rights) so long as it contains a CrossMark logo, the end user license, and a DOI link to the version of record on ScienceDirect. • Retain patent, trademark and other intellectual property rights (including research data). • Proper attribution and credit for the published work.

This agreement deals with the transfer or license of the copyright to Elsevier and authors retain significant rights to use and share their own published articles. Elsevier supports the need for authors to share, disseminate and maximize the impact of their research and these rights, in Elsevier proprietary journals* are defined below:

*Please note that society or third party owned journals may have different publishing agreements. Please see the journal's guide for authors for journal specific copyright information.

**This includes the right for the publisher to make and authorize commercial use, please see "Rights granted to Elsevier" for more details.

RIGHTS GRANTED TO ELSEVIER

For both subscription and open access articles, published in proprietary titles, Elsevier is granted the following rights:

- The exclusive right to publish and distribute an article, and to grant rights to others, including for commercial purposes.
- For open access articles, Elsevier will apply the relevant third party user license where Elsevier publishes the article on its online platforms.
- The right to provide the article in all forms and media so the article can be used on the latest technology even after publication.
- The authority to enforce the rights in the article, on behalf of an author, against third parties, for example in the case of plagiarism or copyright infringement.

BIBLIOGRAPHY

[1] M. Shafique, F. Khalid and S. Rehman, "Intelligent security measures for smart cyber physical systems", 21st Euromicro Conference on Digital System Design (DSD), pp. 280-287, 2018.

[2] D. Lowe, "The ComputerVision Industry," [online] Available: https://www.cs.ubc.ca/~lowe/vision.html

[3] A. M. Lakhwani, K. H. Shah, A. S. Vaghela, D. S. Panchal, and S. R. Rathod, "Review on Basics of Computer Vision and Its Applications," Research & Reviews: Journal of Computational Biology, 6(2), 33-40, 2018.

[4] J. Vongkulbhisal, F. De la Torre, and J. P. Costeira, "Discriminative Optimization: Theory and Applications to Computer Vision," IEEE transactions on pattern analysis and machine intelligence, 41(4), 829-843, 2018.

[5] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey." International journal of computer vision, 128(2), 261-318, 2020.

[6] Z. Q. Zhao, P. Zheng, S. T. Xu, and X. Wu, "Object detection with deep learning: A review," IEEE transactions on neural networks and learning systems, 30(11), 3212-3232, 2019.

[7] J. Han, D. Zhang, G. Cheng, N. Liu, and D. Xu, "Advanced deep-learning techniques for salient and category-specific object detection: a survey," IEEE Signal Processing Magazine, 35(1), 84-100, 2018.

[8] J. Lee, M. Azamfar, J. Singh and S. Siahpour, "Integration of digital twin and deep learning in cyber-physical systems: towards smart manufacturing," IET Collaborative Intelligent Manufacturing, 2(1), 34-36, 2020.

[9] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," In Computer Vision (ICCV), In proceedings of the IEEE International Conference on computer vision, pp. 2980-2988, 2017.

[10] Z. Yin, "Assistive Intelligence (AI): Intelligent Data Analytics Algorithms to Assist Human Experts", 2019.

VITA

Muhammad Monjurul Karim was born in Chittagong, Bangladesh. He received his Bachelor's in Industrial and Production Engineering in 2014 from Bangladesh University of Engineering and Technology. Then he worked at several manufacturing companies and was involved with mathematical optimization in transportation, decision analysis, and systems development. He joined Missouri University of Science and Technology in August 2018 and obtained his master's degree in Systems Engineering in August 2020. He joined Dr. Ruwen Qin's research group to work on his research project, which focused on developing computer vision data based deep learning model for creating complex cyber physical systems. He published two conference papers. He was awarded the runner-up best paper for his outstanding work in one of his papers. Besides publishing papers he also participated in a poster competition to present his research. He also showed his success by achieving the 2nd best poster award in the ISC 2019 poster competition.