01 Jan 2020

A Training Framework of Robotic Operation and Image Analysis for Decision-Making in Bridge Inspection and Preservation

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

Genda Chen
Missouri University of Science and Technology, gchen@mst.edu

Suzanna Long
Missouri University of Science and Technology, longsuz@mst.edu

Zhaozheng Yin
Missouri University of Science and Technology, yinz@mst.edu

et. al. For a complete list of authors, see https://scholarsmine.mst.edu/project_wd-1/1

FINAL REPORT # INSPIRE-0006

GRANT NO: 69A3551747126
GRANT PERIOD: 11/30/16 – 09/30/22
PROJECT PERIOD: 01/01/2018-09/30/2021

Inspecting and Preserving Infrastructure through Robotic Exploration (INSPIRE)

Tier 1 University Transportation Center Sponsored by the Office of the Assistant Secretary for Research and Technology (OST-R)

Project/Report Title:	**A Training Framework of Robotic Operation and Image Analysis for Decision-Making in Bridge Inspection and Preservation**
Consortium Member:	Missouri University of Science and Technology
Principal Investigator:	Ruwen Qin
Co-Principal Investigator(s):	Genda Chen, Suzanna Long, Zhaozheng Yin, Sushil Louis
Report Authors:	Ruwen Qin, Genda Chen, Suzanna Long, Zhaozheng Yin, Sushil Louis, Muhammad Monjurul Karim, Tianyi Zhao

DISCLAIMER

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated in the interest of information exchange. The report is funded, partially or entirely, by a grant from the U.S. Department of Transportation's University Transportation Centers Program. However, the U.S. Government assumes no liability for the contents or use thereof.

TECHNICAL REPORT DOCUMENTATION PAGE

1. Report No. INSPIRE-006	**2. Government Accession No.**	**3. Recipient's Catalog No.**
4. Title and Subtitle A Training Framework of Robotic Operation and Image Analysis for Decision-Making in Bridge Inspection and Preservation		**5. Report Date** July 31, 2020
		6. Performing Organization Code CII
7. Author(s) Ruwen Qin, Genda Chen, Suzanna Long, Zhaozheng Yin, Sushil Louis, Muhammad Monjurul Karim, Tianyi Zhao		**8. Performing Organization Report No.** CII-003
9. Performing Organization Name and Address Center for Intelligent Infrastructure (CII) Missouri University of Science and Technology 211 Engineering Research Laboratory, 500 W. 16th Street Rolla, MO 65409-0810		**10. Work Unit No.**
		11. Contract or Grant No. USDOT:69A3551747126
12. Sponsoring Agency Name and Address Office of the Assistant Secretary for Research and Technology U.S. Department of Transportation 1200 New Jersey Avenue, SE Washington, DC 20590		**13. Type of Report and Period Covered** Final Report Period: 01/01/2018-09/30/2020
		14. Sponsoring Agency Code
15. Supplementary Notes The investigation was conducted in cooperation with the U. S. Department of Transportation.		

16. Abstract

This project aims to create a framework of training engineers and policy makers on robotic operation and image analysis for the inspection and preservation of transportation infrastructure. Specifically, it develops the method for collecting camera-based bridge inspection data and the algorithms for data processing and pattern recognitions; and it creates tools for assisting users on visually analyzing the processed image data and recognized patterns for inspection and preservation decision-making.

The project first developed a Siamese Neural Network to support bridge engineers in analyzing big video data. The network was initially trained by one-shot learning and is fine-tuned iteratively with human in the loop. Bridge engineers define the region of interest initially, then the algorithm retrieves all related regions in the video, which facilitates the engineers to inspect the bridge rather than exhaustively check every frame of the video. Our neural network was evaluated on three bridge inspection videos with promising performances.

Then, the project developed an assistive intelligence system to facilitate inspectors efficiently and accurately detect and segment multiclass bridge elements from inspection videos. A Mask Region-based Convolutional Neural Network was transferred in the studied problem with a small initial training dataset labeled by the inspector. Then, the temporal coherence analysis was used to recover false negative detections of the transferred network. Finally, self-training with a guidance from experienced inspectors was used to iteratively refine the network. Results from a case study have demonstrated that the proposed method uses just a small amount of time and guidance from experienced inspectors to successfully build the assistive intelligence system with an excellent performance.

17. Key Words	**18. Distribution Statement**

Deep Learning, Computer Vision, Big Data Analytics, Bridge Inspection, Worker Assistance	No restrictions. This document is available to the public.	
19. Security Classification (of this report) Unclassified	**20. Security Classification (of this page)** Unclassified	**21. No of Pages** 45

Form DOT F 1700.7 (8-72) Reproduction of form and completed page is authorized.

EXECUTIVE SUMMARY

Inspection of current transportation infrastructure, such as bridges, is an important step towards preservation and rehabilitation of the infrastructure for extending their service lives. With the advancement of mobile robotic technologies, big yet complex inspection video data are rapidly collected. Assisting inspection professionals in retrieving regions of interest from the video data will allow them to concentrate on knowledge intensive tasks and, meanwhile, prepare for the detection and evaluation of defects for comprehensively assessing the bridge condition.

This project first developed a Siamese Neural Network to support bridge engineers in analyzing big video data. The network was initially trained by one-shot learning and is fine-tuned iteratively with human in the loop. Bridge engineers define the region of interest initially, then the algorithm retrieves all related regions in the video, which facilitates the engineers to inspect the bridge rather than exhaustively check every frame of the video. Our neural network was evaluated on three bridge inspection videos with promising performances.

Then, the project developed an assistive intelligence system to facilitate inspectors efficiently and accurately detect and segment multiclass bridge elements from inspection videos. A Mask Region-based Convolutional Neural Network was transferred in the studied problem with a small initial training dataset labeled by the inspector. Then, the temporal coherence analysis was used to recover false negative detections of the transferred network. Finally, self-training with guidance from experienced inspectors was used to iteratively refine the network. Quantitative and qualitative results from a case study have demonstrated that the proposed method uses just a small amount of time and guidance (3.58 hours for labeling 66 images) from experienced inspectors to successfully build the assistive intelligence system with an excellent performance (91.8% precision, 93.6% recall, and 92.7% f1-score).

ACKNOWLEDGMENT

Financial support for this INSPIRE UTC project was 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.

TABLE OF CONTENTS

1. INTRODUCTION

The U.S. roadway transportation system has over 600,000 bridges and 4,300,000 kilometers of public roads [1], which play a critical role in moving passengers and domestic freights. Yet 65% of the roads are rated as the less than good condition, and a quarter of the bridges need significant repair [2]. Rehabilitation, maintenance, and rebuilding efforts are necessary for preserving the transportation infrastructure throughout the United States. Yet manually inspecting and maintaining transportation infrastructure, such as highway bridges, is one of the most costly operations in State Departments of Transportation (DOTs). To address this issue the INSPIRE Center is exploring and developing a remotely-controlled robotic platform that helps with the labor-intensive, risky tasks and allows engineers to focus on knowledge-intensive tasks. Most of bridge engineers are experts in inspection and preservation. Their domain expertise and experiences are valuable, which cannot be easily and quickly acquired by new hires or other workforces. Therefore, an important mission of INSPIRE is to leverage users' (e.g., transportation policy makers and professional engineers) capability of implementing, and interacting with, the robotic platform. This project proposes to fulfill this mission through helping users develop new knowledge and skills of robotic operation, inspection data (such as image data) analysis, and data-driven decision-making. Specifically, this project aims to develop and deliver a training framework for these purposes.

This project explored an approach to engaging bridge inspectors in the development of artificial intelligence algorithms to assist themselves in analyzing bridge inspection video data. Research of this project is focused on the following three aspects: (1) it explores descriptive, predictive, and prescriptive analysis of inspection data acquired by the robotic platform to convert the data (which are complex, large, diverse, and uncertain) into materials and presentation forms that allow users to easily analyze, understand, and utilize in decision-making. (2) It keeps subject matter experts (e.g., engineers and technicians qualified to lead or perform bridge inspection) in the loop and use their expertise to improve the generalization ability of image analysis algorithms. (3) It also creates a training tool for helping users

develop knowledge and skills of analyzing image data to make effective decisions in inspection and preservation.

This project was carried out in two stages. During the first stage of study, the project developed a Siamese Neural Network to support bridge engineers in analyzing big video data. Details of this work are presented in Section 2 of this report. In the second stage, an assistive intelligence system was developed, which engages bridge inspectors in the development of a Convolutional Neural Network for multiclass object detection and segmentation through transfer learning and semi-supervised self-training. Section 3 presents the methodology for developing this assistive system.

2. IMAGE DATA ANALYSIS TO SUPPORT ENGINEERS' DECISION-MAKING

Traditionally, performing bridge inspections in hard-to-access areas is disruptive, difficult and dangerous. In many cases, bridges must be closed to traffic and inspectors must be lifted by heavy equipment. Manual inspection is time-consuming and costly. Using robotics to conduct bridge inspections will be safer, faster, and cheaper. Currently, big data from bridge inspections can be collected from videos recorded with cameras mounted on drones. With a frame rate of 30 frames per second, 108,000 frames can be recorded in one hour. It is a tedious and inefficient process for bridge engineers to watch hours of video footage for bridge inspection.

2.1 Problem description

This project aims to deploy image analysis methodologies to provide decision-making support for bridge inspection through long videos. Fig. 2-1 illustrates the main steps of an automatic retrieval of the region of interest from a long video. An inspector first selects some regions of interest (e.g. joints, beam, surface) in a frame. The image retrieval algorithm developed in this project then finds all related frames in the video. Finally, the collected set of images with localized regions of interest can be evaluated automatically by computer algorithms or verified by inspectors.

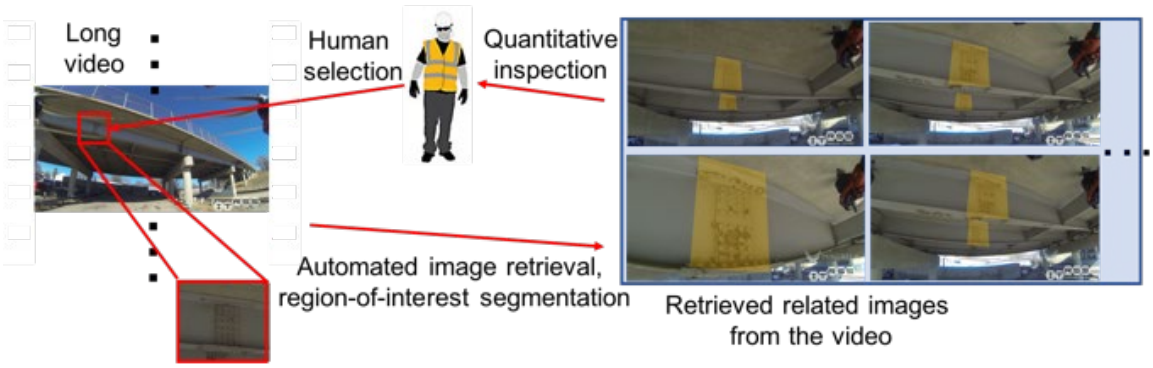


Fig. 2-1 Automatic retrieval of the region of interest based on the initial input by inspectors.

The main challenges include: (1) the viewpoint is changing within a video captured by a camera mounted on a drone, (2) the camera vibration introduced by the drone movement affects the image quality, (3) the regions of interest have different scales in the videos, and (4) the regions to be inspected by bridge engineers may have different visual appearance or types.

A simple template matching or comparing the similarity between hand-crafted features of the query image and reference images may not overcome the previous challenges. Neural networks were used from 1950s to solve the supervised learning problem. At the end of the 20th century, neural networks were applied to the handwriting digital recognition task and achieved superior performance. The neural network method relies on big training data, efficient optimization methods and powerful computation resources. In 2012, deep neural networks [3] were proposed to solve the large-scale image classification problem. Since then, deep neural networks remain the hottest machine learning topic in many industry applications.

In this study, after a bridge engineer selects a target object, we aim at retrieving the similar object from every frame of the video. The goal of our project is to assist engineers with less human effort (e.g., selecting the region of interest by a single image cropping operation). The proposed system has three main contributions: (1) we propose a Siamese neural network that extracts features from the target object patch and video frame using the same network architecture and detects the region of interest by feature similarity comparison; (2) we extend the network into a Multi-scale Siamese Neural Network, which is able to detect the region of interest at multiple scales when the camera on a drone moves away or towards the bridge; (3) since we only have one training sample from the initial selection of the bridge engineer, we leverage the one-shot learning to fine-tune the pre-trained network to the bridge inspection domain, and we propose an iterative approach to further refine the network performance with human-in-the-loop.

2.2 Related work

The Convolutional Neural Network (CNN) is composed of a large amount of neurons organized by multiple convolutional layers. Each layer contains multiple neurons. Each neuron has one convolutional kernel that can perform one particular task (e.g., detecting one particular pattern). From the perspective of transformation function, the first layer transfers the input image to a stack of feature maps. Then the following layers continue to transfer the feature maps to more abstract feature maps. The lower level feature map is more local, for example, edge or texture pattern recognition. The higher level feature map is more abstract, for example, part or object detection. In addition to convolutional layers, CNN has some other layers including pooling layer, normalization layer, fully connected layer, and different connections between layers (e.g., skip connection, dense net, split and merge, multi-scale Inception).

2.3 Methodology

2.3.1 Siamese neural network

The Siamese neural network [4] contains two network architectures which share the same network architecture to compare two images with the same size. We propose a new Siamese neural network that can compare two images with different sizes (i.e., the target object patch and the test image). Our network architecture contains mainly convolutional layers since fully convolutional layers [5] can adapt to input images with different sizes and generate the output with the corresponding size. As shown in Fig. 2-2, the channel number (or the number of convolutional kernels) increases at each layer, while the size (width and height) of the feature maps decreases at each layer. The max pooling layer, whose stride size is 2, decreases the feature map size by 2, as illustrated by one toy example of a single slice of the feature map in Fig.2-3. The size of the feature map generated from a convolutional layer follows the equation:

$$W_O = (W_I + 2 \times pad - ks) / st + 1, \qquad (2\text{-}1)$$

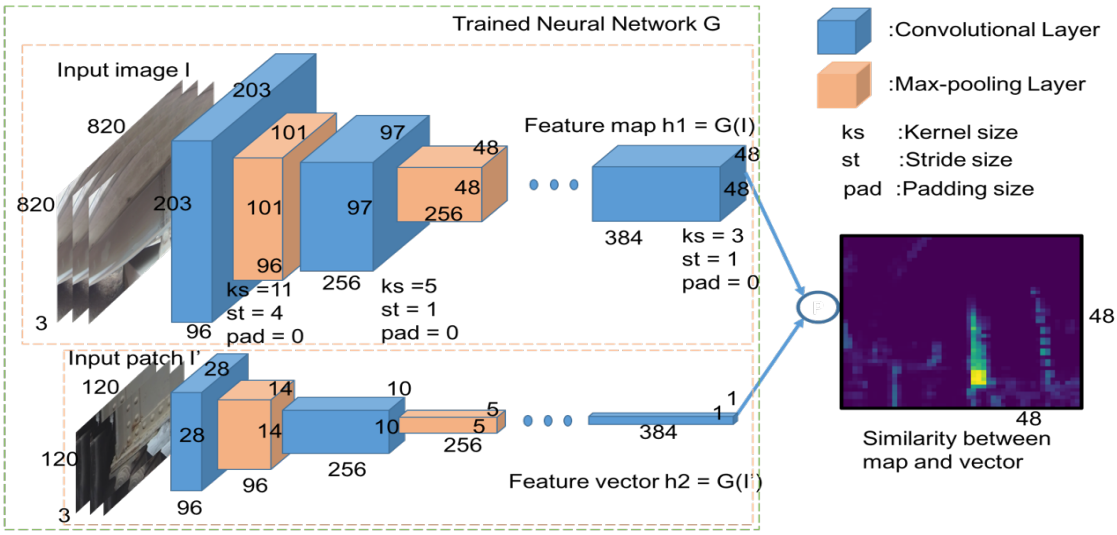


Fig. 2-2 The proposed Siamese neural network.

where W_O and W_I denote the size of output and input, respectively. pad, ks and st denote the number of rows for zero-padding, kernel size and stride size, respectively. For example, the first convolutional layer of the test image in Fig. 2-2 has $pad = 0, ks = 11$, and $st = 4$, so $W_O = \frac{820+2\times0-11}{4} + 1 = 203$.

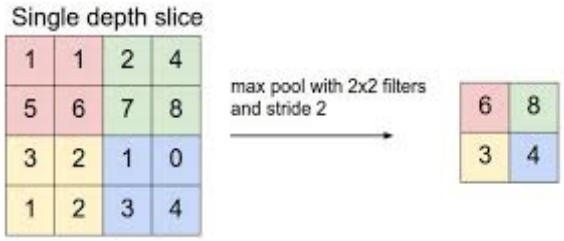


Fig. 2-3 The max-pooling example.

The features from the test image I, $h1 \in R^{W\times H\times K}$, are extracted by Siamese neural network G, where W, H and K denote the width, height and channels of the feature map, respectively ($W = 48, H = 48, K = 384$ in Fig. 2-2). The feature maps of the test image contain the feature vectors at every location, i.e. $h1_{w,h}$ is the feature vector at location (*w,h*). Since the fully convolutional layer can accept different input sizes, given the target object patch, one single feature vector, $h2 \in R^{1\times1\times K}$, is extracted by the same neural network G. The similarity between two feature vectors is measured by

$$P(h1_{w,h}, h2) = Sigmoid\left(\frac{h1_{w,h}\, h2}{|h1_{w,h}||h2|}\right). \quad (2\text{-}2)$$

where $\mathrm{Sigmoid}(x) \triangleq 1/(1+x^{-1})$. The similarity computation between the target object patch and every location in the test image provides a 2D probability map that tells us how likely the object is detected at specific locations in the test image. The two shared network architectures (G) can be trained in an end-to-end manner.

2.3.2 Multi-scale siamese neural network

During the inspection, the camera on a drone may move towards or away from the target area, which changes the object scale continuously. Our image-patch Siamese Neural Network in Fig.2-2 can work well at one scale, but it may fail if the scale changes too much. Thus, we propose a multi-scale Siamese neural network as shown in Fig. 2-4. We up-sample and down-sample the target object patch to a few scales (e.g. $W1 \times H1$ and $W2 \times H2$ in Fig. 2-4). The smaller patch I'' with size $W2 \times H2$ (the camera moves far away from the bridge) will generate the feature vector, $h3 \in R^{1\times1\times K2}$, at the lower level of the network G. The larger patch I' with size $W1 \times H1$ (the camera moves towards the bridge) will get the feature vector, $h2$, at the higher level of the network G. The test image will also be given to the network G and generate the feature maps at different levels. At each level, a 2D probability map is generated, as

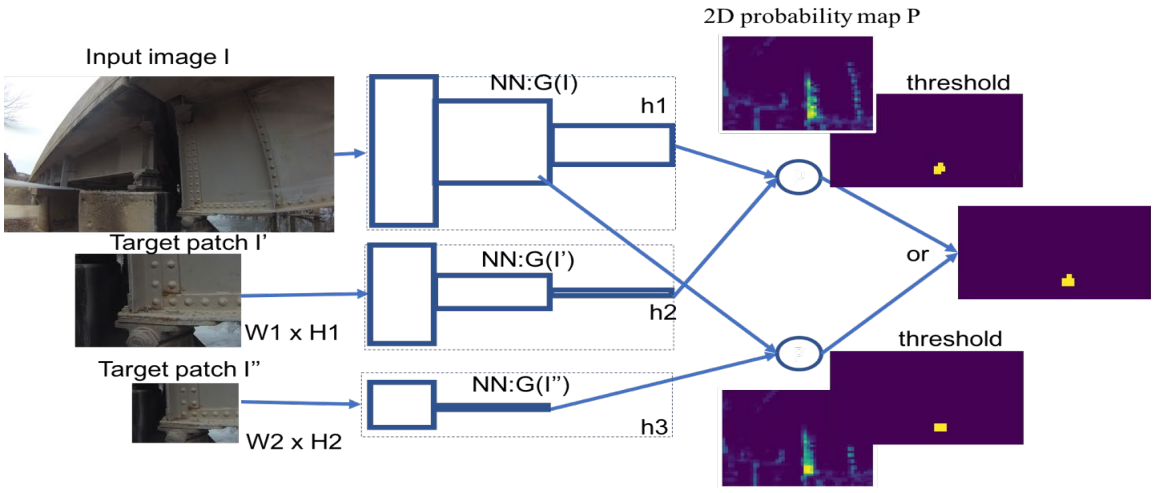


Fig. 2-4 Multi-scale Siamese Neural Network.

described in the previous section. The generated probability map is thresholded as a bitmap. The overall prediction is the union of the thresholded results from all scales.

2.3.3 One-shot learning and fine-tune with human in the loop

The region of interest initially selected by bridge engineers, as a single image patch, is obviously not enough for training a neural network from scratches. Thus, we deploy the pre-trained neural network model AlexNet [3] and then fine-tune the network to fit our bridge inspection project. The AlexNet is trained from the large-scale ImageNet [6] dataset. This dataset contains 1000 categories, and each category contains more than 1000 images. Those 1000 categories don't include the region of interest in our bridge inspection project, but the images in the dataset contain similar texture patterns. To let the network fit the bridge inspection problem much better when we only have one (or a few) training samples, one-shot (or few-shot) learning [7] is intuitively suitable for this situation.

We treat the region of interest detection as a binary classification problem. The region of interest cropped by human is the positive sample, then on the same frame, the other pixels belong to the background. Accordingly, the ground truth map Y is generated based on the positive and negative samples. Let P denote the detected probability map, then the loss to be minimized is a weighted cross-entropy function:

$$\mathrm{Loss} = \sum_{w,h} -\mathrm{Y}_{w,h} \log\left(\mathrm{P}_{w,h}\right) - \alpha\left(1 - \mathrm{Y}_{w,h}\right)\log\left(1 - \mathrm{P}_{w,h}\right). \quad (2\text{-}3)$$

Since there are more negative pixel samples than positive samples, we use weight α to balance the two classes (α is set as 10 in our experiments). The loss function is calculated at each level of our Multi-scale Siamese Neural Network. The overall loss function is a weighted sum of the loss from all levels. After the one-shot training from the initially labeled sample, the detection results over the whole video are generated, which can be visualized as a mask overlaid on the original image as shown in Fig.2-5. Bridge experts can quickly skim the results and identify some false positives and select some correct detections

with large appearance variations. Through the interaction with small human efforts, we can have a little more training data to fine-tune the Multi-scale Siamese Neural Network. The process can be iterated until satisfied. In our experiments, we manually select 15 frames in each iteration, which are added into the fine-tune process. All the frames selected from the current and previous iterations are used for fine-tuning the network in the current iteration.

Fig.2-5 Mask overlays on image

2.4 Experiment

The feature extraction network in our Multi-scale Siamese Neural Network is similar as the first 5 layers of Alexnet. The kernel sizes for the 5 layers are 11, 5, 3, 3, 3. The numbers of kernels for the 5 layers are 96, 256, 384, 384, 256. The stride size is 4, 1, 1, 1, 1. The patch sizes for multi-scale Siamese neural network are 120,70,50, and the feature vectors are extracted from the 5th, 3rd, 2nd convolutional layer. The optimizer is Stochastic Gradient Descent. The learning rate is set as 10^{-1}, then decreases by 10 times until 10^{-4} when the loss doesn't decrease.

Our network is evaluated on three bridge inspection videos. We manually label the ground truth for each frame of the videos. The number of target object (bridge joint in the experiments) in three videos is 181, 56, 70, respectively. For each video, we perform two iterations of human interaction. Table 2-1 summarizes the evaluation results, from which we observe that the iteratively fine-tuning can improve the performance gradually. Due to the large appearance variation, there are still some miss detections

and false alarms by our network using the initial one-shot training. More training samples will definitely help overcome this problem and we leave this as our future work. Some qualitative results are shown in Fig. 2-6. The first column is the hard prediction, which is thresholded form the soft prediction (the second column of Fig. 2-6). The last column is the original test image with the bounding box representing the detected area.

Table. 2-1 The detection performance on 3 videos. One shot learning column shows the results generated by the model trained by the first image patch only. Iteration 1 shows the results after the first round of human interaction with 15 frames. Iteration 2 shows the results after the second round of human interaction with 30 frames which include 15 frames from the first round. 'prec' denotes the precision. 'f1' denotes the f1 score.

Videos	One-shot learning			Iteration 1			Iteration 2		
	prec	recall	f1	prec	recall	f1	prec	recall	f1
Video 1 (7m:25s)	.333	.448	.382	.730	.589	.652	.877	.514	.673
Video 2 (5m:42s)	.376	.75	.501	.693	.964	.806	.678	1.0	.809
Video 3 (2m:26s)	.348	.712	.467	.773	.787	.78	.818	.863	.84

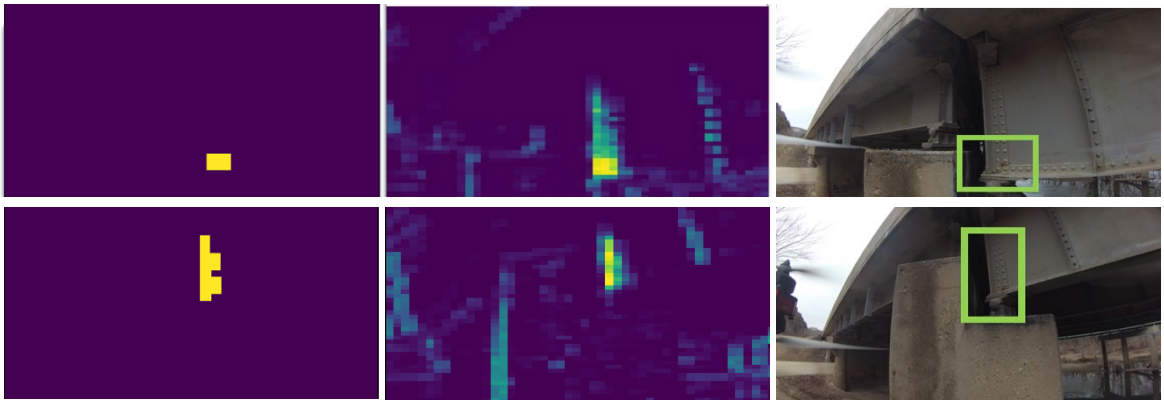


Fig. 2-6 Hard prediction, soft prediction and detected bounding box on the test image.

2.5 Conclusion

In this study, we aim to develop image analysis algorithms to provide decision-making support for bridge inspection through long videos. Our proposed algorithms include: (1) image-patch Siamese neural

network; (2) multi-scale Siamese neural network; (3) one-shot learning and iteratively fine-tuning the pre-trained network with human-in-the-loop.

3. AN ASSISTIVE INTELLIGENCE SYSTEM FOR FACILITATING BRIDGE ENGINEERS IN ANALYZING INSPECTION VIDEO DATA

The U.S National Bridge Inventory has over 600,000 highway bridges. 39% of these bridges are over 50 years old, and almost 9% are structurally deficient and require significant repair [8]. Rehabilitation, maintenance, and rebuilding efforts are necessary for preserving the transportation infrastructure throughout the United States. For example, National Bridge Inspection Standards require that each bridge should be inspected every two years to ensure that there are no cracks, rust, or other damages [9]. The conventional bridge inspection requires a crew of inspection professionals, heavy equipment with a lifting capability, the access to dangerous heights, and the closure of the road during the time of inspection. These make the bridge inspection one of the most dangerous and costly operations in the state Departments of Transportation.

Research has taken place to develop safer and more efficient bridge inspection methods. Some adopted a completely manual approach for the bridge routine inspection, which requires a large number of inspector hours [10-11] and inspection results vary largely among inspectors. To reduce inspector hours and the bias of human inspection, methods to automate the bridge inspection have been developed. For example, a laser scanning method [12-14] and the photogrammetry [15-16] have been used for collecting inspection data. The reliability of these methods is dependent on the expertise of operators and influenced by the weather condition.

Recently, mobile robots such as Unmanned Aerial Vehicles (UAVs) have been proven to be very helpful in some dangerous, dull, or dirty applications [17-18]. This data collection method attempts to reduce or eliminate the labor-intensive onsite inspection process and allows inspectors to assess bridges from a safer location. However, analyzing the collected inspection video data is still a very challenging task for practitioners. On one hand, there could be hours of video data that need to be inspected and analyzed

for each individual bridge of inspection. On the other hand, bridge inspection videos captured by mobile inspection robots are mainly images of complex scenes, wherein a bridge of various structural elements is mixed with a cluttered background. Assisting inspectors in analyzing the big, complex video data is greatly needed.

Provided with the big data from inspection, researchers have been exploring deep learning techniques for the defect detection [19-28]. Many of these detection models take close-up images of a single component in a nearly uniform testing background where defects are relatively large and straightforward to recognize. Besides, ratings of bridges need to be provided by a comprehensive bridge assessment that evaluates the impact of defects on specific elements of bridges [29-30]. This requires to spatially relate detected defects with the bridge elements where the defects are located. The above-mentioned needs suggest that an important step of assisting inspectors is to detect regions of interest and retrieve relevant frames from inspection videos. After that, defect evaluation and interpretation can take place. Per our knowledge, no work has taken place to detect and segment important bridge elements from video data captured by mobile robots such as UAVs.

3.1 Problem description

Deep learning methods for detecting and segmenting multiclass objects from images are well developed. However, performing this task on videos captured by mobile inspection platforms are difficult due to various reasons such as largely varied views of objects in videos, motion blur, partial or full occlusion, illumination variation, background variation, and so on. So far, some studies [31-34] have reported the success of detecting objects from videos, for example, utilizing the temporal information of objects. But the additional computational cost is nontrivial. The high accuracy of deep learning models for multiclass object detection and segmentation relies on large-scale dense annotations for model training. Yet annotating a huge amount of training data is not only labor-intensive but expensive as it may need the

knowledge of domain experts [35]. To truly assist them in their jobs, the burden of data annotation should not be completely passed to inspectors. Efforts that domain experts contribute to the deep learning model development must be well controlled and best utilized.

This report proposes a cost-effective method to create an assistive intelligence system for detecting and segmenting multiclass objects from inspection video data captured by UAVs. The system is not an artificial intelligence model isolated from users. Instead, inspectors provide inputs and guide the system development to assure it quickly converges to a satisfactory tool for assisting themselves in analyzing the videos of any intended bridge of inspection. Filling the gaps identified in this report, the proposed method has anticipated contributions in three folds: (i) a quick transfer of an existing deep learning network to the task of detecting and segmenting multiclass bridge elements from complex inspection video data, (ii) the use of a lightweight temporal coherence method that does not require any repetitive action to boost up the network's ability of object detection, and (iii) the development of a semi-supervised self-training method that keeps human-in-the-loop to efficiently refine the deep neural network iteratively.

3.2 Related work

Computer vision based techniques have been used in many studies for defect detection in structural elements. For instance, studies in [19, 21, 24] focused on detecting single type of defects in structural components using computer vision. These techniques require both pre- and post-processing of images, thus being time consuming. Then, with the rapid advancement of data collection and computing technologies, deep learning has diffused in the computer vision based inspection data analysis. For example, researchers used Convolutional Neural Network (CNN) based methods [22-23, 36-37] for multiclass target detection. Dealing with the characteristic change of objects among different scenes is still very challenging with these techniques. Besides, this approach comes with a high computational cost because the CNN classifier must be applied many times for every single sliding window in each image. It

has been noticed that most defect detection models [22-23, 36-37] take close-up images of a single component in a nearly uniform testing background where defects are relatively large and straightforward to recognize.

To improve the efficiency in detecting and locating multiclass objects, a region-based CNN (R-CNN) has been introduced [38]. R-CNN uses the selective search [39] to generate region proposals to find objects in an image. To make it faster the same author proposed faster R-CNN [40] that uses region proposal network (RPN). The faster R-CNN offers improvements in both speed and accuracy over its predecessors through shared computation and the use of neural networks to propose regions. Then, Mask R-CNN [41], an extension of faster R-CNN, was developed to perform both bounding box regression and pixel-level segmentation simultaneously.

Above-mentioned R-CNN and Mask R-CNN models work well in detecting objects in static images. Yet, results may not be consistent when they process video data. Therefore, the temporal coherence of an object in successive frames has been introduced to address the issue of inconsistent detection [31-33], wherein the tubelet and optical flow are used to propagate features from one frame to another. However, approaches to temporal coherence analysis in the literature are computationally expensive due to the requirement for repeated motion estimation and feature propagation. Seq-NMS [34] has a modification only in the post-processing phase and, thus, it is faster than others. However, seq-NMS tends to increase the volume of false positive detection because it neither puts any penalty on the false positive detection nor adds additional constraints to prevent its occurrence.

Creating a deep learning model usually requires a huge amount of annotated data for model training. The manual annotation of data is not only a costly process but often prone to errors. To overcome this issue, some researchers have used transfer learning [42] in structural damage detection [25-28]. With transfer learning, they used a relatively small dataset of structural elements to refine a deep learning model, which

reduced the training time while keeping a good performance. Another way to tackle this annotation problem is to use Semi-Supervised Learning (SSL). This training method requires some labeled and some unlabeled training data. Papandreou et al. [43] developed a method that requires a small number of strongly annotated images and a large number of weakly annotated images for training. They used an expectation maximization method to generate the pixel-level annotation from weakly annotated training data. Chang et al. [44] used labeled and unlabeled training data for feature selection. Mittal et al. [45] proposed an approach that relies on adversarial training with a feature matching loss to learn from unlabeled images. On the other hand, some researchers use self-training wrapper based semi-supervised method [46] which starts training a network with only a few annotated samples and then let the network automatically annotate more training samples. This technique has been applied to a variety of image/video processing applications [47-48] to reduce the effort of human annotation. While most SSL methods save annotation efforts compared to supervised learning, the performance is still not good enough. The primary reason for this challenge is the quality of the automatically annotated data. The inclusion of some samples mislabeled by the network itself may sharply deteriorate the training process.

3.2 Methodology

The proposed approach to creating an assistive intelligent system for multiclass bridge element detection and segmentation is illustrated in Fig. 3-1, which is built on three major methods. At first, a pre-trained Mask R-CNN has been chosen. A small set of initial training data, which are annotated by the inspector, is used to fine-tune the network to adapt the network to the multiclass bridge element detection and segmentation task. Then, the trained network is applied to an unlabeled test dataset for generating detection and segmentation results. If the network does not provide a satisfactory performance, temporal coherence analysis is applied to recover some false negative detections and improve the result. Then a small set of representative samples is selected from the detection and segmentation result to create

additional training data. The data is then split into two subsets, one subset has been automatically annotated by the trained Mask R-CNN and the other subset is manually re-annotated by the inspector. The additional training data along with the initial small training dataset is then used to refine the Mask R-CNN. The performance of the refined network will again be assessed using the remainder of unlabeled test dataset which has excluded the portion used for training. This iterative semi-supervised self-training process will be continued until a satisfied performance is obtained. The proposed Semi-Supervised Learning (SSL) is a self-training approach but it let the inspector to annotate the initial training dataset and the selected samples that the current network failed to detect. Through learning from its weakness, the performance of the network increases quickly after a few iterations.

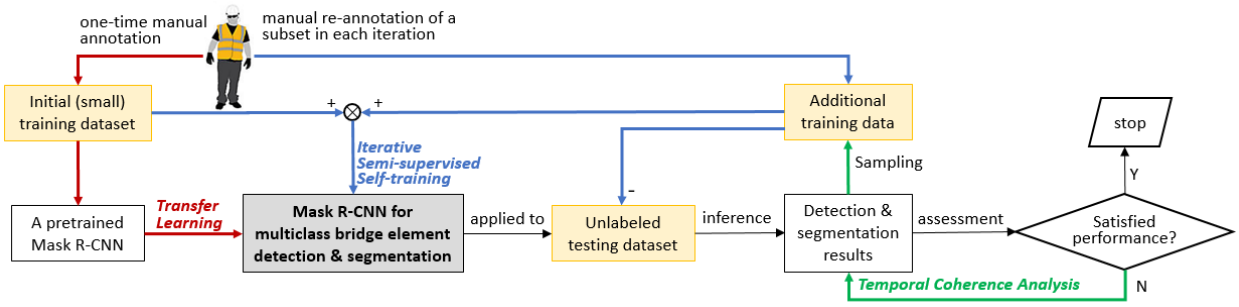


Fig. 3-1 Overview of the proposed method to develop an assistive intelligence system for bridge element detection and segmentation.

3.2.1 Adapting the deep neural network to a new task through transfer learning

This study chose Mask R-CNN, a type of R-CNN that performs the region segmentation at the pixel-level, as the tool for detecting and segmenting bridge elements from inspection video data. Fig. 3-2 illustrates the structure of Mask R-CNN. The backbone of the network is a feature extractor that generates the feature map of each input image. A region proposal network (RPN) creates proposal boxes named anchors and predicts the possibility of an anchor being a bridge element. Then the RPN ranks anchors and proposes those most likely containing bridge elements, which are termed RoIs. A layer named Region of Interests

Align (RoIAlign) extracts the region of interests (RoIs) from the feature map, aligns them with the input image, and converts them into fixed-size region feature maps. The fixed-size feature maps of RoIs are fed into two independent branches: the network head branch that performs the classification and bounding box generation, and the mask branch that independently generates instance masks. Interested readers can refer to [41] for details.

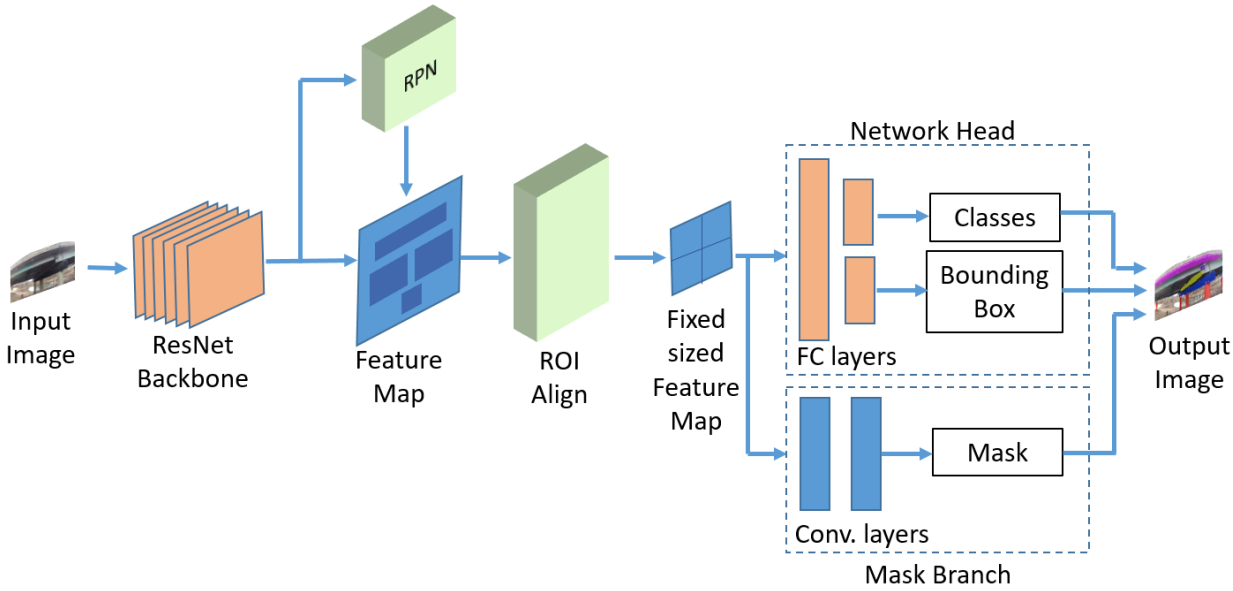


Fig. 3-2 The architecture of Mask R-CNN

Training Mask R-CNN from scratch requires a large volume of annotated data to achieve a satisfied prediction accuracy. The bridge element detection and segmentation is a new task of image analysis that does not have a large volume of annotated data for model training. To obtain high quality annotated data for this task requires the knowledge of professionals in the domain of study. Only bridge inspectors are confident in annotating bridge elements from the inspection videos. In this study, transfer learning is used to tackle this challenge, which improves learning in the new task through transferring knowledge from a related task that has already been learned [42].

The Mask R-CNN in this study was initialized by adopting the ResNet-50 feature extractor [49] whose

weights have been pre-trained on the Microsoft COCO dataset consisting of more than 120,000 labeled images and around 1.5 millions of object instances in 80 categories [35]. Then, transfer learning was used to adapt this feature extractor to the specific setting of bridge inspection. Specifically, the ResNet-50 was fine-tuned using a small set of training data (T0) mainly collected from the intended bridges of inspection. Details of the fine-tuning process are presented in Section 3.3.1-1.

3.2.2 Temporal coherence analysis for recovering false negative detections

Mask R-CNN is a static image detector in that it processes individual images independently. When it is applied to frames of a video stream, false negative detections are likely to happen due to sudden scale changes, occlusion, or motion blur. This study used the temporal coherence information of objects in successive frames to recover false negative detections.

Consider a video clip that consists of a series of N frames, indexed by i. In each frame the detector returns M_i objects, indexed by j. An object in a frame is highly likely present in its neighboring frames within a range of displacement. 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 $C_{i,j} = (x_{i,j}, y_{i,j})$. In p frames, $C_{i,j}$ may shift to a surrounding pixel within a spatial displacement of $p\Delta d$ where Δd is the maximum displacement between two consecutive frames. The selection of Δd depends upon two factors, the speed of the UAV and the distance between objects and the camera. Δd is proportional to the speed of the UAV and inversely proportional to the distance between objects and the camera. A Δd value of 60 pixel was found to be appropriate in this study. Fig. 3-3 illustrates an example wherein a joint of the bridge in frame $i - 4$ is also shown in the succeeding four frames but with displacements.

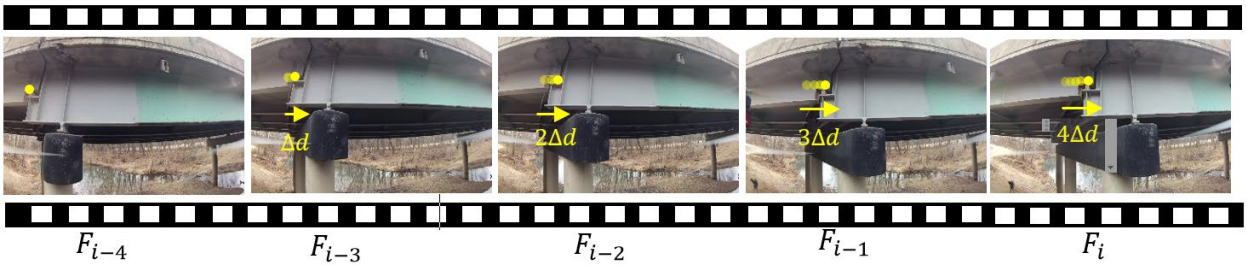


Fig. 3-3 An illustration of spatial displacements

The algorithm of temporal coherence analysis for recovering false negative detections is summarized as the pseudo-code in Algorithm 1. The detection threshold is set as a range $[t_l;\ t_u]$. An object with a detection score within this range is possibly a false negative detection. Let $S_{i,j}$ denote the detection score for object $o_{i,j}$. The detector immediately returns a positive detection if $S_{i,j} \geq t_u$ and will not detect any object if $S_{i,j} < t_l$. Let O_i be the set of confidently detected objects in frame i. The detection score and the center location of these objects, $\{(S_{i,j}, C_{i,j})|o_{i,j} \in O_i\}$ are the temporal coherence information for analyzing the succeeding k frames. If $t_l \leq S_{i,j} < t_u$, the weakly detected object $o_{i,j}$ is checked by referring to a pair of preceding successive frames up to $k-1$ times, starting from the nearest pair (frames $i-1$ and $i-2$) to the farthest pair (frames $i-k+1$ and $i-k$). If an object of the same class as $o_{i,j}$ is found in both frames $i-1$ and $i-2$ (i.e., there exists $o_{i-1,j'} \in O_{i-1}$ and $o_{i-2,j''} \in O_{i-2}$ such that $o_{i-1,j'} = o_{i-2,j''} = o_{i,j}$), and the spatial displacements of $o_{i,j}$ from $o_{i-1,j'}$ and $o_{i-2,j''}$ are small, within Δd and $2\Delta d$, respectively, this weakly detected object is assumed a false negative detection. It is recovered through adding it to O_i and updating its detection score to be the average score of $S_{i-1,j'}$ and $S_{i-2,j''}$. Otherwise, $o_{i,j}$ is searched in O_{i-2} and O_{i-3} to determine if it is a false negative detection that can be recovered. This search will continue as needed. If $o_{i,j}$ is not found to be a positive detection with confidence in the neighboring frames after $k-1$ times of temporal coherence analysis, it is reported as a false positive detection in the target frame and will be eliminated from the detection list. It is noticed that searching an object in pairs of successive frames minimizes the risk of progressively propagating false

positive detection to succeeding frames. Using a pair of frames as reference instead of referring to a single frame will make the temporal coherence rules stricter. Therefore, any false positive detection in a single frame is less likely propagated to the target frame. A value for the parameter k was selected with a consideration of both the UAV speed and camera speed in taking videos for inspection. A k value of 4 was found to be suitable in this study.

Algorithm 1 Temporal Coherence Analysis for Recovering False Negative Detections

// N: the number of video frames;
// M_i : the number of objects in frame i;
// O_i : the set of confidently detected objects in frame i;
// $S_{i,j}$: the detection score of the jth object in frame i;
// $C_{i,j}$: the center location of the jth object in frame i;
// t_l and t_u: detection thresholds;
// k: the number of frames that have stored temporal coherence information of detected objects.

for $i = 1$ *to* N **do**
  for $j = 1$ *to* M_i **do**
    if $S_{i,j} \geq t_u$ **then**
      add object $o_{i,j}$ to 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 $q = 1, 2, ...(k-1)$ **do**
        if $\exists\, o_{i-q,j'} \in O_{i-q}$ & $o_{i-q-1,j''} \in O_{i-q-1}$, $\ni\, o_{i,j} = o_{i-q,j'} = o_{i-q-1,j''}$ & $||C_{i,j} - C_{i-q,j'}||_2 \leq q\Delta d$ & $||C_{i,j} - C_{i-q-1,j''}||_2 \leq (q+1)\Delta d$, **then**
          let $S_{i,j} = (S_{i-q,j'} + S_{i-q-1,j''})/2$,
          add object $o_{i,j}$ to the set O_i with its $C_{i,j}$ and updated $S_{i,j}$,
          break
        end if
      end for
    end if
    Eliminate the low score $(S_{i,j} < t_u)$ object $o_{i,j}$ from the detection list.
  end for
end for

3.2.3 Refining the network through iterative semi-supervised learning

After transfer learning has initialized the Mask R-CNN for the task of bridge inspection, the network needs to be further refined, for example, by adding additional training data annotated by the inspector. If the refined network has not reached a satisfied performance, the refining process will continue. To lower the cost of data annotation and, meanwhile, maintain a good quality of the training data, the study chose Semi-Supervised Learning (SSL) with a guidance from the inspector for the continuous refining of the network, which uses a mix of inspector-annotated data and model-annotated data. That is, a self-training wrapper based semi-supervised learning method is proposed as the following. In each iteration, the trained deep neural network (NN) is applied to a set of unlabeled data to be labeled automatically. Using the temporal context (TC) information, hard samples are collected from this newly created labeled data. A representative subset of the hard samples is identified and added to the training dataset to refine the network. Before this subset is added to the training dataset, a portion of it are manually re-annotated by the inspector to guide the learning process of the SSL. This process continues iteratively until the model performance converges to a steady state. The iterative approach allows for gradually seeking a satisfied performance, with a minimal human effort. On one hand, the proposed iterative SSL method addresses the difficulty of SSL in determining the optimal amount of additional training data and the optimal fraction of inspector-annotated data. One the other hand, it selects additional training data from hard samples so that the deep NN can efficiently improve itself by learning from its weakness.

The iterative re-training approach with SSL is delineated in Fig. 3-4 and the complete algorithm is further summarized as the pseudocode in Algorithm 2. Let l denote the index of iterations. A small training dataset T_0 is created to transfer the Mask R-CNN. Then, the network is applied to an unlabeled dataset, V. If the performance of the network is not satisfied, a portion of the segmentation result is sampled as the additional training data, denoted as S_l. For the initial iteration (i.e., $l = 0$), this S_l is sampled from

the entire segmentation result to guide the network. In each iteration, the sampled data S_l is eliminated from V for further assessment of future networks. For the following iterations (i.e., $l > 0$), the SSL algorithm differentiates hard samples from easy samples in V.

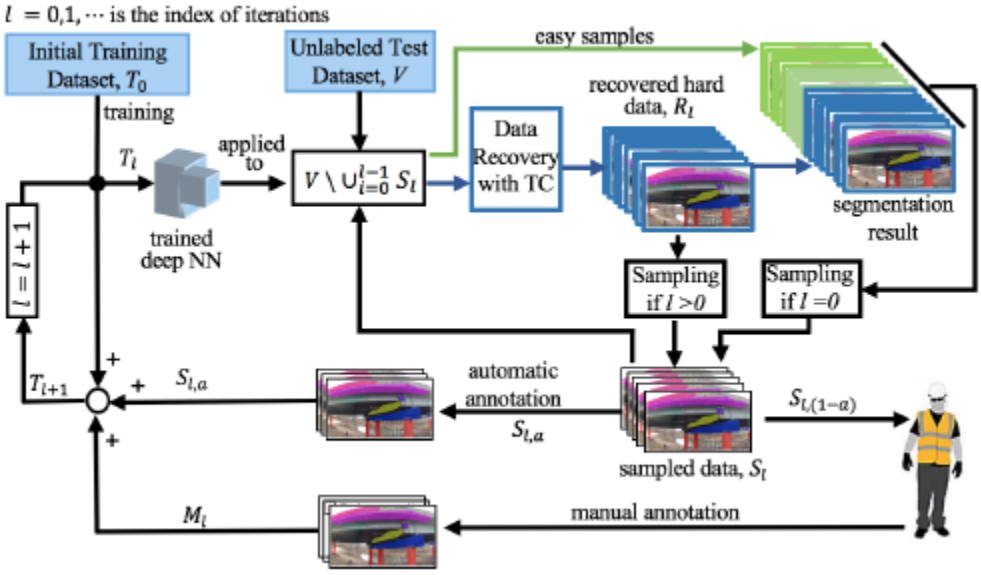


Fig. 3-4 Schematic diagram of the iterative semi-supervised learning for refining the Mask R-CNN model

Easy samples are segmented by the network with relatively high reliability whereas hard samples,R_l, which contain a variety of situations when objects are difficult to detect, are recovered by the developed temporal coherence method described in section-III-B. The sample S_l (for $l > 0$) is selected only from the hard recovered data R_l as per section-3.2.3(1). S_l is divided into two mutually exclusive and collectively exhaustive subsets, $S_{l,\alpha}$ and $S_{l,1-\alpha}$where α and 1 - α indicate their sizes in proportion to S_l. $S_{l,\alpha}$ has been automatically annotated by the trained network in testing and directly added to the training dataset. The inspector re-examines the remainder of S_l and corrects the false detection, if any, before adding the inspector-annotated data M_l to the training dataset. That is, at the end of the lth iteration the training dataset is updated per Eq. (3-1):

$$T_{l+1} = T_l \cup M_l \cup S_{l,\alpha} \tag{3-1}$$

The network is re-trained using the updated training dataset and the segmentation result from the l^{th} iteration is assessed. If the termination criterion has been met, the fine-tuning process is terminated. Otherwise, the SSL continues refining the network. The termination criterion of the iterative process is subject to the decision of users. This study chose to terminate the iterative process when the performance starts to converge.

Algorithm 2 Iterative Fine-Tuning the Network with SSL

// l: index of iteration;
// T_l: the training dataset for iteration l;
// V: unlabeled dataset for the SSL;
// R_l: the recovered hard dataset from temporal coherence analysis;
// S_l: a subset of R_l, which is sampled based on the sampling method SP(s);
// $S_{l,\alpha}$: a fraction of S_l in the size of α automatically annotated by the trained network;
// $S_{l,(1-\alpha)}$: a fraction of S_l in the size of $1-\alpha$ to be manually annotated by the inspector;
// M_l: the data re-annotated by the inspector and added to the training dataset in iteration l.
for $l \geq 0$ **do**
  Fine-tune the network with T_l,
  obtain R_l through the temporal coherence analysis,
  break if the performance meets the requirement,
  sample S_l from the segmentation result if $l = 0$,
  sample S_l from R_l if $l > 0$, per Section III-C1,
  split S_l into two mutually exclusive parts,
  manually re-annotate $S_{l,(1-\alpha)}$ to obtain M_l,
  $T_{l+1} = T_l \cup M_l \cup S_{l,\alpha}$,
  increase α per Section III-C2,
  break if $\alpha \geq 1$,
end for

This iterative process has two designs: 1) the method for sampling S_l from R_l and, 2) the way of determining the fraction of S_l to be re-examined by the inspector.

1) Skip sampling method, SP(s): Consecutive frames of a video are similar and, therefore, sampling a

portion of frames that are evenly distributed on the timeline would be sufficient for representing the video. Therefore, this study samples S_l from R_l, for any iteration l, according to a skip sampling strategy SP(s) that samples a frame and then skips s frames. The choice of a value for s needs to consider the UAV speed and the camera speed. The testing dataset V is a time series of N_v frames. I_{SP} is a $1 \times N_v$ indicator vector of binary variables that define frames to be sampled according to SP(s); that is,

$$I_{SP}(n) = 1, \tag{3-2}$$

For $n = 1, 1 + (s + 1), \dots, 1 + (s + 1)\lfloor N_v/(s + 1) \rfloor$. $I_{R,l}$ is also a $1 \times N_v$ indicator vector of binary variables that identify the frames recovered by the temporal coherence analysis. The Hadamard product of $I_{R,l}$ and I_{SP} yields the vector $I_{S,l}$,

$$I_{S,l} = I_{R,l} \circ I_{SP} \tag{3-3}$$

which identifies the frames to be sampled from R_l according to SP(s) for forming S_l.

2) Regulating the amount of human guidance in iterative SSL:

A fraction of the dataset S_l from any iteration is automatically annotated by the trained network. The initial performance of the neural network is not high and data mislabeled by the network are present in Sl. Through examining a fraction of S_l and correcting mislabeled data, the experienced inspector guides the network to quickly learn new features. $S_{l,\alpha}$ is the fraction of S_l which is added to the training dataset without further human annotation. The human guidance can be gradually reduced as the network starts to learn well by itself. Therefore, the fraction of automatically annotated data $S_{l,\alpha}$ can be gradually increased over iterations. Choosing the initial value of α for $S_{l,\alpha}$ is also critical as it regulates the amount of mislabeled data that enter the training dataset. Determining an optimal selection of α for the iterative SSL is a research problem but going beyond the scope of this work.

3.3 A case study and the results

This section illustrates the development and evaluation of the proposed assistive system for a bridge of inspection. Findings from this study are discussed.

3.3.1 Implementation details of system development

1) The Data: A mobile camera attached with a customized multi-copter UAV was used to capture videos of the bridge of inspection. The average speed of the UAV is 20 miles per hour (mph). The frame rate of the camera is 30 frames per second (fps) and the image resolution is 3840 x 2160 in pixel. A dataset D, which is an inspection video of 4440 images, was used to develop and evaluate the assistive system for the bridge. The initial training dataset T_0 contains 40 images, with 18 images from D and 22 images from the inspection of other bridges. Choosing some images of other bridges adds helpful data variation to the initial training dataset. In total the initial training dataset contains 482 objects with class labels, which are from 10 different classes of bridge elements interested to inspect. The ten object classes are barrier, slab, pier, pier cap, diaphragm, joint, bearing, pier wall, bracket, and rivet, illustrated in Fig. 3-5.

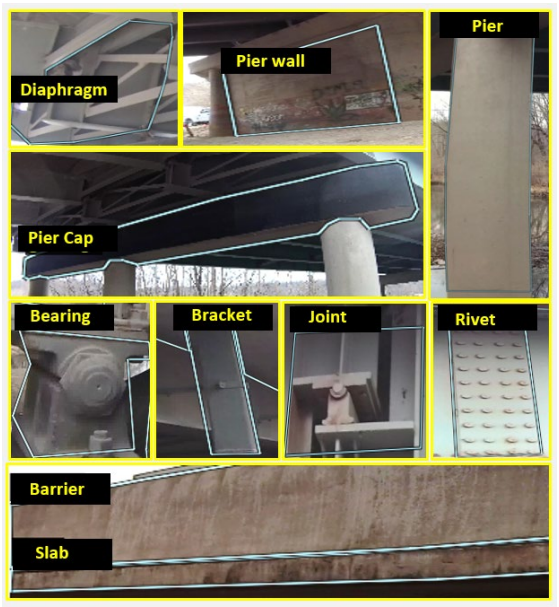


Fig. 3-5 Sample images with corresponding pixel-level object polygon with labels.

This study used the image annotation tool VGG Image annotator [50] to annotate labels of the objects and give pixel-level coordinates to those objects. An unannotated dataset V that comprises 670 images from the dataset D was particularly created for implementing the iterative SSL method. V contains 5,916 objects from the 10 classes. A test dataset T_S has been created to evaluate the model performance after each iteration. This dataset has 212 images with 1,872 objects.

2) Initial adaptation: The proposed method was implemented by extending an existing implementation of Mask R-CNN by Matterport Inc. [51]. Training and testing were performed using two Nvidia Tesla V100 GPUs with 32GB of memory. The pre-trained ResNet-50 feature extractor was fine-tuned using the initial training dataset T0. The network head and the mask head (see Fig. 2) were trained for 30 epochs while keeping all the parameters of the previous layers 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 of 0.001 and a batch size of 4 were used in this training process that took about 21 minutes to complete. According to Algorithm 2, after the Mask R-CNN is transferred to have an initial adaption to the task of bridge inspection, V is annotated by this network. Considering the performance of the initially adapted network, 8 images, which are about 1% images in V, are selected and re-annotated by the inspector and added to the initial training dataset T_0, becoming T_1, the training dataset for the next iteration. These 8 images are excluded from V for further iterations. Letting the inspector to check a small amount of the segmentation result of the initially adapted network in accordance of the performance is a practical approach to controlling the quality of training dataset.

3) Inference and iterative refining: Table 3-1 summarizes the iterative training process for fine-tuning the deep NN with iterative SSL method using human guidance. Refining the network for the 1st iteration of the SSL method was initiated with the last epoch of the previous iteration (i.e., the transfer learning) and continued for 20 more epochs. Then, the remainder of dataset V, V n S0, is annotated by the refined

network. This study considered 0.5 as the lower boundary of detection threshold tl and 0.9 as the upper boundary of detection threshold t_u, which were found to minimize the volume of false detection based on numerical experiments. Temporal coherence analysis (TCA) is applied to the samples of $V \backslash S_0$ which contain objects with detection scores between 0.5 and 0.9 to recover false negative detections. The temporal coherence analysis in the 1st iteration was able to recover 113 frames, and 37 frames of these were sampled according to SP(2), the sampling strategy considered by this study. In this study α was 70% in the 1st iteration, which means the inspector re-annotated 30% of S1 before adding them to the training dataset. In 2nd iteration, the network was refined using the updated training dataset T_2 and then it was used to evaluate $V \backslash S_0 \cup S_1$. TCA recovered 79 images and 33 images were sampled and added to the training dataset. α was reduced to 20% of S2 and the inspector re-annotated 7 images out of the 33 before adding them to the training dataset. The iterative process was terminated after the 3rd iteration when the improvement was saturated.

Table 3-1 Data sizes (#images) in transfer learning and iterative semi-supervised learning

	Transfer learning	Iterative SSL		
l: index of iterations	0	1	2	3
T_l: training dataset	40	48	85	118
R_l: recovered hard data samples	-	113	79	50
S_l: a sampled subset of R_l	-	37	33	
α: % of S_l for automatic annotation	-	70%	80%	
$S_{l,\alpha}$:: automatically annotated data	-	26	26	
M_l: manually annotated data	8	11	7	

3.3.2 Quantitative results

In this study, experiments were conducted to demonstrate the cost-effectiveness of the proposed method. The developed network performs two different tasks: object detection and instance segmentation. To evaluate the performance of the developed network on these two tasks, different standard evaluation matrices were used in this study. In addition to measuring the detection and segmentation accuracy, two more experiments were conducted to evaluate the efficiency and the generalization ability of the proposed method. This study also compared the proposed approach with the state-of-the-art method to assess the achieved improvement.

1) Object detection results: To evaluate the performance of object detection with the developed deep network, three standard evaluation matrices were used in this study:

- precision: it counts the number of correct predictions out of the total number of predictions;
- recall: it counts the number of correct predictions out of total number of ground-truth objects;
- f1-score: it is the harmonic mean of precision and recall.

This study used the Intersection over Union (IoU) to determine whether a predicted object can be considered as a correct detection. The IoU is the intersection between the predicted bounding box and the ground truth bounding box over the union of them. The ability of the network to correctly detect objects was evaluated on a range of IoU threshold values from 0.1 to 0.9 at a step of 0.1. The precision, recall, and f1-score from evaluating the test dataset T_s are summarized in Table 3-2. Given an IoU threshold value in the range of [0.1, 0.5], the Mask R-CNN that was initially fine-tuned to adapt to the bridge of inspection achieved the precision from 80.3% to 87.5%, the recall from 74.4% to 81.0%, and f1-score from 77.2% to 84.1%. The performance indicates the fine-tuned network demonstrated some adaptability to the new task, but the amount of false negative detection is non-negligible. The performance of the network has not reached a satisfied level.

Therefore, the network was iteratively refined using the proposed SSL method, seeking further improvement. After being re-trained in the 1st iteration, the recall was effectively increased by 15%, approximately. For example, when the IoU threshold value is 0.5, the precision increases from 80.3% to 81.7%, and recall becomes 90.3% from 74.4%, yielding a 85.8% f1-score after the 1st iteration. The changes indicate that M0, the additional small set of manually annotated samples, effectively improves the ability to correctly detect more objects. The performance of the network has converged after being further refined for additional two iterations, reaching 91.8% precision, 93.6% recall, and 92.7% f1-score at the IoU threshold value 0.5. The iterative SSL has effectively brought the performance of the network to a satisfied level. As the IoU threshold value decreases gradually from 0.5 to 0.1 at a step size of 0.1, the evaluation becomes less conservative. Consequently, less false negative detections are rendered by the network but maybe more false positive detections. On the other hand, selecting a higher IoU threshold value makes the evaluation more conservative. As it increases gradually from 0.5 to 0.9 at a step size of 0.1, the f1-score is diminishing, signifying the reduction of both precision and recall values. In this application setting, false positive detections are less concerned than false negative detections. This is because the inspector will retrieve and analyze frames that contain detected and segmented objects that s/he wants to inspect. Therefore, false positive detections can be found and eliminated by the inspector. But, false negatives are more critical because inspectors cannot overlook any potential damages. From the analysis above it can be inferred that, this iterative SSL method is very applicable to the development of the proposed assistive system for detecting bridge elements from inspection videos.

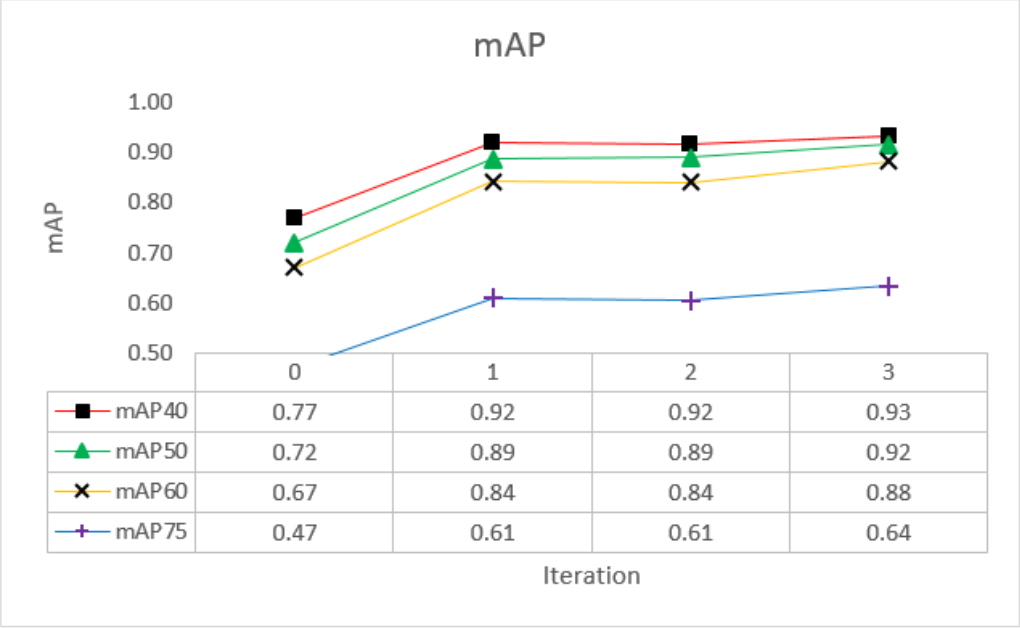
Another important observation from Table 3-2 is the relationship between the IoU threshold value and the recall value during the iterative process of network fine-tuning. When the IoU threshold value increases from 0.1 to 0.5, a variation of a recall value within a range of 6.6% has been observed after the initial adaption through transfer learning. However, this variation reduces in each successive iteration. For

example, after the third iteration this variation reduces to 1.6%. Precision and recall values at any of the IoU threshold values increase over iteration and reach the maximum after the 3rd iteration. For example, at the IoU threshold value 0.5, f1 score increases about 8.6%, 4.6% and 2.3%, respectively, from their previous iteration. This means the network learns new features from each iteration and gradually moves toward the learning limit. The improvement rate is diminishing during the iterative process. When the IoU threshold value continues increasing from 0.5 and onward, the recall value in any iteration drops rapidly and has been less than 10% when the IoU threshold value is 0.9. Moreover, the increasing trend of the recall value over iterations quickly slows down at the IoU threshold value 0.6. When the IoU threshold value is greater than 0.6, the increasing trend of recall over iterations is rapidly flatted out and becomes a decreasing trend. The reason for this sharp decrease of recall value along with the increase of the IoU threshold value is that the network considers a detected object as a true positive detection only if the overlap between the ground truth and the bounding box of the detected object is very high, which increases the amount of false negative detections and decreases the amount of true positive detections.

Table 3-2 Performance (%) at different iterations

		Transfer learning	Iterative SSL		
	Iterations, l	0	1	2	3
IoU					
0.1	Precision	87.5	86.3	94.0	93.4
	Recall	81.0	95.4	93.5	95.3
	F1-Score	84.1	90.6	93.8	94.4
0.2	Precision	87.1	85.7	93.9	93.5
	Recall	80.7	94.8	93.3	95.3
	F1-Score	83.4	90.0	93.6	94.4
0.3	Precision	86.8	85.2	93.7	93.4
	Recall	80.3	94.2	93.2	95.2
	F1-Score	83.4	89.5	93.4	94.3
0.4	Precision	84.6	84.2	93.2	93.1
	Recall	78.3	93.1	92.7	94.9
	F1-Score	81.3	88.5	93.0	94.0
0.5	Precision	80.3	81.7	90.7	91.8
	Recall	74.4	90.3	90.1	93.6
	F1-Score	77.2	85.8	90.4	92.7
0.6	Precision	75.9	77.4	85.7	88.5
	Recall	70.2	85.6	85.1	90.2
	F1-Score	73.0	81.3	85.4	89.3
0.7	Precision	65.4	66.6	74.6	78.1
	Recall	60.5	73.6	74.2	79.6
	F1-Score	60.5	73.6	74.2	79.6
0.8	Precision	43.7	43.8	50.1	49.0
	Recall	40.5	48.5	49.8	49.9
	F1-Score	42.1	46.0	49.9	49.5
0.9	Precision	6.3	3.0	6.7	4.6
	Recall	5.9	3.3	6.7	4.7
	F1-Score	6.1	3.1	6.7	4.7

2) Instance segmentation results: The study also evaluated the ability of the proposed system to segment instances from inspection videos using the metric named mean Average Precision (mAP). In this study, mAP is the averaged precision of the ten individual classes. Fig. 3-6 shows the curve of mAP value during the iterative process of fine-tuning the network at four levels of mask IoU threshold value, wherein the x-axis represents the number of iterations and the y-axis represents the mAP value. The plot shows the mAP curve at the mask IoU threshold value 0.4 is an upwarding curve on the top of other curves. The mAP value at the end of the iterative SSL reaches 93.1%. When the mask IoU threshold value increases to 0.5, the curve just drops slightly and the shape of the curve has no change. The mAP value at the end of the iterative SSL reaches 91.5%. However, with a larger mask IoU threshold value such as 0.6 or 0.75, the mAP curve drops to a lower position. This is because the amount of true positive detections at larger IoU threshold values is low although the total number of correctly segmented objects increases over iterations.



	0	1	2	3
mAP40	0.77	0.92	0.92	0.93
mAP50	0.72	0.89	0.89	0.92
mAP60	0.67	0.84	0.84	0.88
mAP75	0.47	0.61	0.61	0.64

Fig. 3-6 mAP over iterations at different IoU threshold values

3) Efficiency of transfer learning: To demonstrate the high cost-effectiveness of transfer learning for the initial adaption, this study trained a Mask R-CNN from the scratch using 144 annotated images. Results of the comparison are summarized in Table 3-3. The first 600 epochs for training the network from scratch took 13.2 hours, and the network has not reached a converged performance by then. This network gave a poor performance (precision: 32.3%, recall: 18.3%, and f1: 23.4% with an IoU threshold value 0.5) when it was tested on the dataset Ts. The experiment clearly demonstrates the network requires a huge number of training samples to be trained from the scratch, which is infeasible for developing the desired bridge inspection tool because of labeled data scarcity. The proposed transfer learning used only 40 annotated images as the training dataset and took only 20 minutes to transfer the capability of an existing Mask R-CNN in multiclass object detection and segmentation to the new task with bridge elements. The performance of the transferred Mask R-CNN has a much better result (precision: 80.3%, recall: 74.4%, f1: 77.2% with an IoU of 0.5). The comparison summarized in Table 3-3 demonstrates that transfer learning reduces the training time by at least 95% and tremendously improved the performance of the network.

Table 3-3 Cost-effectiveness of transfer learning in comparison with training from scratch

Method	Training time	Precision	Recall	F1
Training from scratch	13.2 hrs	32.3%	18.3%	23.4%
Transfer learning	0.33 hrs	80.3%	74.4%	77.2%

4) Generalization capability of the proposed approach: To assess if the developed network is applicable to other bridges, the study used it to detect and segment the same ten classes of bridge elements of another two bridges, named bridges A and B. Correspondingly, the bridge that has had a network developed for is named bridge C. When the network built for bridge C was applied to bridges A and B, the performance of it is comparable to that from analyzing bridge C without the human guided SSL. The trained network achieves 76.8% of precision, 73.0% of recall, and 74.8% of f1-score for bridge A; and for

bridge B it accomplishes 61.2% of precision, 60.0% of recall, and 60.6% of f1-score. Bridge A is more similar to bridge C than bridge B. Therefore, the network trained for bridge C performs better in analyzing the inspection data of bridge A than bridge B.

5) Comparison with a State-of-the art method: This study compared the proposed approach (i.e., transfer learning plus iterative SSL) to the Mask R-CNN directly adapted with transfer learning, from the perspectives of annotation time, training time, inference speed, and accuracy. To show the reliance of the performance of the transferred Mask R-CNN on the volume of training dataset, this study measured the performance of the Mask R-CNN after independently transferred in with four random training datasets: 0.5% (22 images), 1% (44 images), 5% (220 images) and 10% (440 images) of the images in Dataset D. Results from the comparison are summarized in Fig. 3-7 and Table 3-4. It can be observed that the transfer learning by itself can improve the performance of the network, but the improvement is at a rapidly increasing cost of annotation time. Transferring the Mask R-CNN with 440 annotated images took 1430 minutes (i.e., 23.8 hours) for data annotation and 66 minutes for training. This network achieves 89.7% of precision, 92.3% of recall, and 91.0% of f1 score, close to the performance of the proposed approach in this study. The proposed approach reduces the annotation time by 85% with a comparable training time (only 6 minutes longer), and it achieves a better performance (91.8% of precision, 93.6% of recall, and 92.7% of f1). The inference time for both Mask R-CNN and the proposed approach is 0.55 seconds per frame.

6) A comparison between with and without the assistive system: An inspector takes an average of 3.25 minutes to detect and manually segment bridge elements in a full image. Although the manual work by inspectors can achieve near 100% accuracy, the time cost makes it almost infeasible considering the big volume of inspection images collected rapidly. The developed assistive system can finish the same job in very high accuracy but with only 0.55 seconds per image, which is 350 times faster than the manual

approach. The impact of the improved work efficiency is tremendous because a real-world task usually requires analyzing hundreds of thousands of images. Meanwhile, the accuracy achieved by an inspector will drop when the inspector is manually reviewing inspection images for a long period of time due to human factors related issues (e.g., the loss of attention and the accumulation of fatigue).

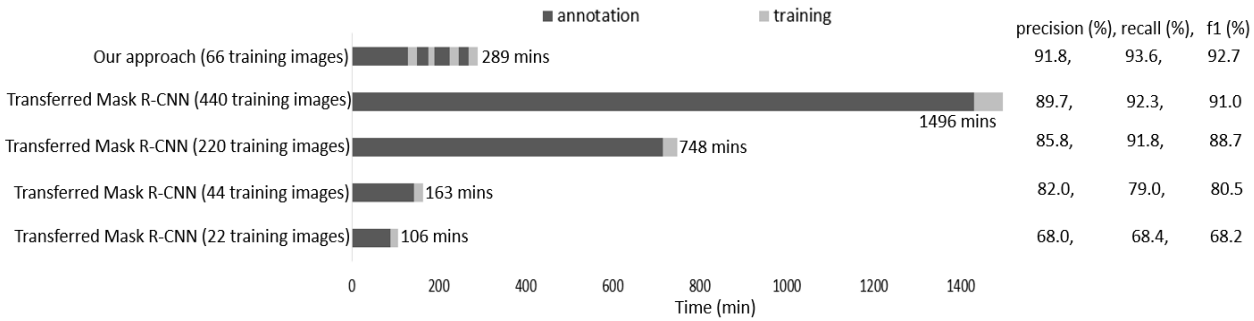


Fig. 3-7 A comparison of model development time

Table 3-4 Comparison of the proposed approach to directly transferring a Mask R-CNN with various volumes of training dataset

Method	No. of manually annotated images	Annotation time (min)	Training time (min)	Inference time (sec/frame)	Precision (%)	Recall (%)	F1 (%)
Mask R-CNN	27	88	18	0.55	68.0	68.4	68.2
Mask R-CNN	44	143	20	0.55	82.0	79.0	80.5
Mask R-CNN	220	715	33	0.55	85.8	91.8	88.7
Mask R-CNN	440	1430	66	0.55	89.7	92.3	91.0
Our approach	66	215	72	0.55	91.8	93.6	92.7

In the real-world implementation, the developed system will detect and segment bridge elements from the inspection image data. Then, the inspector retrieves the elements of interests from the large pool of video frames and evaluate damages or other defects associated with them. The developed system assists inspectors in that it reduces the human effort in searching and finding the needed data so that inspectors can focus on knowledge-intensive tasks.

3.3.3 Qualitative results

The developed system was tested on the inspection dataset D. Some qualitative examples selected from the testing result are illustrated below.

1) An illustrative example of temporal coherence analysis: Fig. 3-8 presents an example wherein the temporal coherence analysis improves the model performance by eliminating false negative detections. In the first row of Fig.8, the single-image based detector correctly detects the diaphragm in the 1st and the 4th frames, however, fails to detect it in the 2nd and the 3rd frames. The second row is the result after applying the temporal coherence analysis, which shows that the diaphragm is correctly segmented in all of the four frames. Note that false negative detections are more severe than false positive detections in the task of bridge inspection. Because, false positive detections rendered by the deep NN can be re-checked by the inspectors but false negative detection ignored by the network will not have such an opportunity. Therefore, effectively reducing false negative detections is particularly more important for bridge inspection.

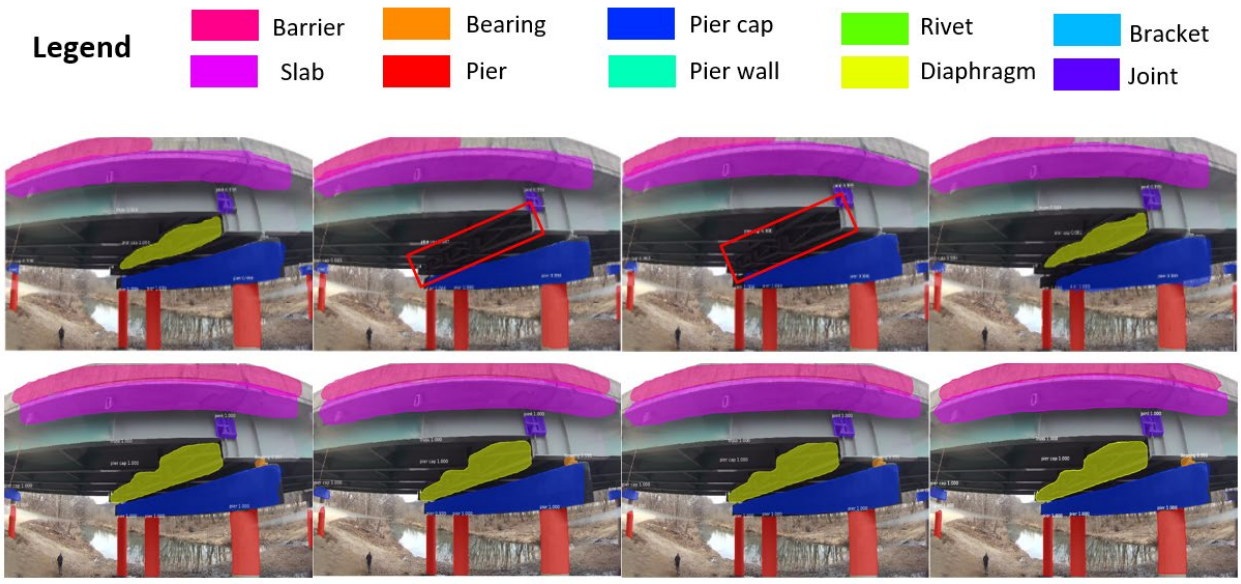


Fig. 3-8 An example of temporal coherence analysis

2) Representative examples of successful detection and segmentation: Fig.3-9 illustrates some representative examples of successful detection and segmentation of bridge elements by the developed system. The first column of Fig 9 are examples that a partial joint in different scales in images is detected and segmented correctly from all three images. The second column illustrates the capability of detecting and segmenting a rivet from various viewpoints. The network successfully detects and segments the rivet in a low light condition, as the second figure in column two illustrates. The third column comprises successful examples of segmenting the rivet at a wide range of scale variations. The last column shows that the developed network is successful in detecting and segmenting multiple objects at various distances in complex scenes.

Fig 3-9. Examples of successful detection and segmentation.

3) Representative examples of false negative detection: The red bounding boxes in Fig. 3-10 represent false negative detections by the developed network. The diaphragm in the 1st frame and the pier cap in the 2nd frame are difficult to recognize because of the dark illumination level. In the 3rd frame the

developed network fails to detect the barrier due to the high exposure. The rivet in the third frame is not detected because of its similar appearance with the background. We consider external illumination sources on UAVs and image contrast enhancement techniques as potential solutions to overcome these challenges.

Fig. 3-10 Examples of false negative detection

3.4 Conclusions

This study presents the development of an assistive intelligence system for supporting bridge inspectors in detecting and segmenting multiclass bridge elements from big, complex video data captured by mobile robots such as UAVs. With a small initial training dataset, a Mask R-CNN pre-trained on a large public dataset was transferred in the studied problem. Then, the temporal coherence analysis was used to boost the performance of the transferred network through recovering the false negative detection, which just requires storing a small set of temporal context information for a few frames. Therefore, this method adds a nearly negligible additional computation load during the inference compared to other methods based on motion guidance. Semi-Supervised Learning with human guidance was also developed to leverage the recovered hard data to refine the network in an alternative manner, which quickly brought its performance to a satisfied level.

A case study of bridge inspection has demonstrated that the proposed method for system development uses very limited amount of time (like a few hours) from domain experts to achieve a high performance

in detecting and segmenting multiclass objects from big, complex inspection videos. For example, the developed system has achieved around 94% of precision, 93% of recall, and 92% mAP50 when the IoU threshold value is 0.5. The study has revealed that having sufficient guidance from experienced bridge inspectors, particularly in early iterations of semi-supervised learning for refining the network, is critical for maintaining the quality of the training dataset. The amount of human guidance can be gradually reduced as the network is becoming more reliable in performing its tasks.

This work has identified rooms for improvement. While the developed system is able to achieve a high performance with a small amount of human hours, and computation time for training the network. However, there is still scope for improving the testing speed to have real time inference capability. An important future work is to improve the inference speed. Another future work could include contextual information and spatial correlation among objects to further minimize the false negative detection.

4. RECOMMENDATIONS DEVELOPED AS A RESULT OF THE PROJECT

This project explored the engagement of bridge professionals in developing deep learning algorithms for analyzing inspection video data captured by mobile inspection platforms. Firstly, a Siamese Neural Network was initially trained by one-short learning and then was fine-tuned iteratively. Then, an assistive intelligence system was built, which was transferred in to the problem of study and iteratively improved through semi-supervised self-training. Bridge professionals were kept in the loop in developing both models. Assessment results and comparative studies have demonstrated the cost-effectiveness of the proposed approach.

Based on the current project, recommendations are made as the following:

- No deep learning models are directly applicable to any tasks and, therefore, it is important to adapt models to different tasks to achieve satisfied performance. Transfer learning and iterative boosting are useful methods that let the algorithms learn new features of a task from a small amount of data of the task, thus adapting to that task.
- Keep-human-in-the loop is an important method to leverage human intelligence into the artificial intelligence algorithms. This should be achieved through a collaborative approach. That is, algorithms provide humans with its performance so that humans can figure out the weakness and provide inputs (e.g., additional training dataset annotated by humans) to the algorithms for improvement.
- In this project, model adaptability and the collaboration between artificial intelligence and human experts were integrated together as a solution for developing assistive intelligence that takes care of time-consuming, boring tasks and let humans focus on knowledge-intensive tasks. This will be a new style of work for future bridge professionals.

5. REFERENCES

[1] Bureau of Transportation Statistics. https://www.bts.gov/. Accessed on Sept 16 2017.

[2] US Department of Transportation. Beyond traffic 2045: Trend and choices. 2015.

[3] A. Krizhevsky, I. Sutskever, and G.E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in Neural Information Processing Systems. 2012.

[4] G. Koch, R. Zemel, and R. Salakhutdinov. "Siamese neural networks for one-shot image recognition." ICML Deep Learning Workshop. Vol. 2. 2015.

[5] J. Long, E. Shelhamer, and T. Darrell. "Fully convolutional networks for semantic segmentation." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

[6] J. Deng, W. Dong, R. Socher, J.J. Li, and F.-F Li. "Imagenet: A large-scale hierarchical image database." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2009.

[7] Q. Cai, Y. Pan, T. Yao, C. Yan and T. Mei. "Memory matching networks for one-shot image recognition." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.

[8] ASCE, "2017 infrastructure report card." ASCE Reston, VA, 2017.

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

[10] D. V. J´auregui and K. R. White, "Implementation of virtual reality in routine bridge inspection," Transportation Research Record, vol. 1827, no. 1, pp. 29–35, 2003.

[11] J. Bauer, N. S¨underhauf, and P. Protzel, "Comparing several implementations of two recently published feature detectors," IFAC Proceedings Volumes, vol. 40, no. 15, pp. 143–148, 2007.

[12] T. Ditto, J. Knapp, and S. Biro, "3d inspection microscope using holographic primary objective," Proceedings of SPIE - The International Society for Optical Engineering, vol. 7432, pp. 24–, 08 2009.

[13] 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, vol. 9, no. 1, pp. 568–601, 2009.

[14] E. J. Jaselskis, Z. Gao, and R. C. Walters, "Improving transportation projects using laser scanning," Journal of Construction Engineering and Management, vol. 131, no. 3, pp. 377–384, 2005.

[15] Z. Zhu and I. Brilakis, "Comparison of civil infrastructure optical-based spatial data acquisition techniques," Journal of Computing in Civil Engineering, vol. 23, no. 3, pp. 170–177, 2009.

[16] P. Arias, J. Armesto, D. Di-Capua, R. Gonz´alez-Drigo, H. Lorenzo, and V. Perez-Gracia, "Digital photogrammetry, gpr and computational analysis of structural damages in a mediaeval bridge," Engineering Failure Analysis, vol. 14, no. 8, pp. 1444–1457, 2007.

[17] D. Weatherington, "Unmanned aerial vehicles roadmap: 2002-2027," Office of the Secretary of Defense, 2002.

[18] S. Minaeian, J. Liu, and Y.-J. Son, "Vision-based target detection and localization via a team of cooperative uav and ugvs," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 46, no. 7, pp. 1005–1016, 2015.

[19] E. Zalama, J. G´omez-Garc´ıa-Bermejo, R. Medina, and J. Llamas, "Road crack detection using visual features extracted by gabor filters," Computer-Aided Civil and Infrastructure Engineering, vol. 29, no. 5, pp. 342–358, 2014.

[20] X. Tao, D. Zhang, Z. Wang, X. Liu, H. Zhang, and D. Xu, "Detection of power line insulator defects using aerial images analyzed with convolutional neural networks," IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2018.

[21] Y.-J. Cha, K. You, and W. Choi, "Vision-based detection of loosened bolts using the hough transform and support vector machines," Automation in Construction, vol. 71, pp. 181–188, 2016.

[22] N. S. Gulgec, M. Tak´aˇc, and S. N. Pakzad, "Structural damage detection using convolutional neural networks," in Model Validation and Uncertainty Quantification, Volume 3. Springer, 2017, pp. 331–337.

[23] Y.-J. Cha, W. Choi, and O. B¨uy¨uk¨ozt¨urk, "Deep learning-based crack damage detection using convolutional neural networks," Computer- Aided Civil and Infrastructure Engineering, vol. 32, no. 5, pp. 361–378, 2017.

[24] J. Gan, J. Wang, H. Yu, Q. Li, and Z. Shi, "Online rail surface inspection utilizing spatial consistency and continuity," IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2018.

[25] C. Zhang, C. Chang, and M. Jamshidi, "Bridge damage detection using single-stage detector and field inspection images," arXiv preprint arXiv:1812.10590, 2018.

[26] Y. Gao and K. M. Mosalam, "Deep transfer learning for image-based structural damage recognition," Computer-Aided Civil and Infrastructure Engineering, vol. 33, no. 9, pp. 748–768, 2018.

[27] C. Zhang and C.-C. Chang, "Surface damage detection for concrete bridges using single-stage convolutional neural networks," in Health Monitoring of Structural and Biological Systems XIII, vol. 10972. International Society for Optics and Photonics, 2019, p. 109722E.

[28] K. Gopalakrishnan, H. Gholami, A. Vidyadharan, A. Choudhary, and A. Agrawal, "Crack damage detection in unmanned aerial vehicle images of civil infrastructure using pre-trained deep learning model," Int. J. Traffic Transp. Eng, vol. 8, pp. 1–14, 2018.

[29] A. A. of State Highway and T. Officials, Manual for Bridge Element Inspection, 2nd ed., 2019.

[30] Z. Zhu, S. German, and I. Brilakis, "Detection of large-scale concrete columns for automated bridge inspection," Automation in Construction, vol. 19, no. 8, pp. 1047–1055, 2010.

[31] K. Kang, H. Li, J. Yan, X. Zeng, B. Yang, T. Xiao, C. Zhang, Z. Wang, R. Wang, X. Wang et al., "T-cnn: Tubelets with convolutional neural networks for object detection from videos," IEEE Transactions on Circuits and Systems for Video Technology, vol. 28, no. 10, pp. 2896–2907, 2018.

[32] 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, 2017, pp. 2349–2358.

[33] 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, 2017, pp. 408–417.

[34] [27] W. Han, P. Khorrami, T. L. Paine, P. Ramachandran, M. Babaeizadeh, H. Shi, J. Li, S. Yan, and T. S. Huang, "Seq-nms for video object detection," arXiv preprint arXiv:1602.08465, 2016.

[35] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Doll´ar, and C. L. Zitnick, "Microsoft coco: Common objects in context," in European Conference on Computer Vision. Springer, 2014, pp. 740–755.

[36] K. Makantasis, E. Protopapadakis, A. Doulamis, N. Doulamis, and C. Loupos, "Deep convolutional neural networks for efficient vision based tunnel inspection," in 2015 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP). IEEE, 2015, pp. 335–342.

[37] L. Yang, B. Li, W. Li, Z. Liu, G. Yang, and J. Xiao, "Deep concrete inspection using unmanned aerial vehicle towards cssc database," in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2017, pp. 24–8.

[38] 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, 2014, pp. 580–587.

[39] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," International Journal of Computer Vision, vol. 104, no. 2, pp. 154–171, 2013.

[40] R. Girshick, "Fast r-cnn," in Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440–1448.

[41] K. He, G. Gkioxari, P. Doll´ar, and R. Girshick, "Mask r-cnn," in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2961–2969.

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

[43] G. Papandreou, L.-C. Chen, K. P. Murphy, and A. L. Yuille, "Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation," in Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1742–1750.

[44] X. Chang and Y. Yang, "Semisupervised feature analysis by mining correlations among multiple tasks," IEEE Transactions on Neural Networks and Learning Systems, vol. 28, no. 10, pp. 2294–2305, 2016.

[45] S. Mittal, M. Tatarchenko, and T. Brox, "Semi-supervised semantic segmentation with high-and low-level consistency," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019.

[46] I. Triguero, S. Garc´ıa, and F. Herrera, "Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study," Knowledge and Information systems, vol. 42, no. 2, pp. 245–284, 2015.

[47] W.-C. Hung, Y.-H. Tsai, Y.-T. Liou, Y.-Y. Lin, and M.-H. Yang, "Adversarial learning for semi-supervised semantic segmentation," arXiv preprint arXiv:1802.07934, 2018.

[48] I. Misra, A. Shrivastava, and M. Hebert, "Watch and learn: Semi-supervised learning for object detectors from video," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3593–3602.

[49] 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, 2016, pp. 770–778.

[50] A. Dutta and A. Zisserman, “The vgg image annotator (via),” arXiv preprint arXiv:1904.10699, 2019.

[51] W. Abdulla, “Mask r-cnn for object detection and instance segmentation on keras and tensorflow,” 2017. [Online]. Available: https://github.com/matterport/Maskn RCNN