COCO-Bridge: Structural Detail Data Set for Bridge Inspections

Eric Bianchi[1]; Amos Lynn Abbott, Ph.D.[2]; Pratap Tokekar, Ph.D.[3]; and Matthew Hebdon, Ph.D., P.E., M.ASCE[4]

Abstract: The purpose of this research is to propose a means to address two issues faced by unmanned aerial vehicles (UAVs) during bridge inspection. The first issue is that UAVs have a notoriously difficult time operating near bridges. This is because of the potential for the navigation signal to be lost between the operator and the UAV. Therefore, there is a push to automate or semiautomate the UAV inspection process. One way to improve automation is by improving UAVs' ability to contextualize their environment through object detection and object avoidance. The second issue is that, to the best of the authors' knowledge, no method has been developed to automatically contextualize detected defects to a structural bridge detail during or after UAV flight. Significant research has been conducted on UAVs' ability to detect defects, like cracks and corrosion. However, detecting the presence of a defect alone does not contextualize its significance or help with an inspector's job to rate specific structural bridge details. This paper outlines a use case for a data set and model to detect critical structural bridge details, providing context and vision for enhancing the autonomous UAV bridge inspection process. Identifying these structural bridge details that require inspection may assist an UAV in path planning and object avoidance in GPS-denied environments. The detection of structural details adds an ability to contextualize defect detection and localize issues to a bridge detail. This also has implications for providing cues to inspectors, in real time, on defect-susceptible areas while UAVs are in flight. The image data set, Common Objects in Context for bridge inspection (COCO-Bridge), for UAV object detection was collected and then trained using deep learning techniques. This data set consists of 774 images and over 2,500 object instances to detect 4 structural bridge details: bearings, cover plate terminations, gusset plate connections, and out-of-plane stiffeners. These details were chosen because they either must be rated by an inspector or checked because they are prone to failure. Methods to economize the predictive capabilities of the model through image augmentation were investigated to extend the performance of the training images. It was concluded that for this domain of data, structural bridge detail images, the mean average precision, and F1 score performance were improved by mirroring the training images along their *y*-axis. The outcome of this paper was an open-source annotated data set, which can be used in computer vision applications for visual inspection, growing the capabilities of artificial intelligence in structural engineering. **DOI: 10.1061/(ASCE)CP.1943-5487.0000949.**

Introduction

The 2017 infrastructure report card from the American Society of Civil Engineers found that over 40% of bridges were 50 years or older (Otero 2015). Monitoring infrastructure and preventive maintenance save money and continue to grow in importance as bridges age. Additionally, technology has been advancing to improve the bridge inspection process. In particular, unmanned aerial vehicles (UAVs) hold promise as a method to assist inspectors. They may help by avoiding traffic obstruction and accessing hard-to-reach areas that may otherwise require special equipment. In research, UAVs have been used to create 3D point clouds of bridges (Gillins et al. 2018) and identify instances of change over time (Hallermann et al. 2015). UAVs can raise the standards for inspection by retrieving useful bridge data and collecting new bridge data, which was not feasible or economical using traditional bridge inspection practices. However, UAVs present challenges of their own. This is because of the variable wind speeds around bridges, UAV battery life, loss of Global Positioning System (GPS) signal as a result of the sheer size of a structure, and compass disruption stemming from the amount of steel in bridges. Automating portions of the UAV flight, or operating semiautonomously, is one way to mitigate the dangers of crashing an UAV in GPS-denied environments and compass signal loss. Semiautonomous UAV flight means that the inspector or the UAV operator would still be able to take back control of the UAV if necessary. An UAV's ability to operate semiautonomously during an inspection process has the potential to make inspections more efficient, effective, and repeatable. The proposed image-based data set offers a computer vision method to facilitate UAV semiautonomous inspection processes by identifying structural details. Structural bridge details are bridge elements that an inspector is responsible for evaluating following the *Bridge Inspector's Reference Manual* (BIRM) of the Federal Highway Administration (FHWA) (Ryan et al. 2012).

The outcomes of detecting structural bridge details would improve several inspection aspects, including data recording and contextualizing structural defects, instance mapping, object avoidance, and path planning, in the successful autonomous or semiautonomous inspection of a bridge. Identifying structural details contextualizes structural defects found on or around detected structural

[1]Assistant Professor, Charles E. Via Jr. Dept. of Civil and Environmental Engineering, Virginia Tech, Blacksburg, VA 24060 (corresponding author). Email: beric7@vt.edu

[2]Professor, Bradley Dept. of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA 24060. Email: abbott@vt.edu

[3]Assistant Professor, Dept. of Computer Sciences, Univ. of Maryland, College Park, MD 20742. Email: tokekar@umd.edu

[4]Assistant Professor, Charles E. Via Jr. Dept. of Civil and Environmental Engineering, Virginia Tech, Blacksburg, VA 24060. ORCID: https://orcid.org/0000-0002-9115-0279. Email: mhebdon@vt.edu

Note. This manuscript was submitted on February 27, 2020; approved on September 2, 2020; published online on January 30, 2021. Discussion period open until June 30, 2021; separate discussions must be submitted for individual papers. This paper is part of the ***Journal of Computing in Civil Engineering***, © ASCE, ISSN 0887-3801.

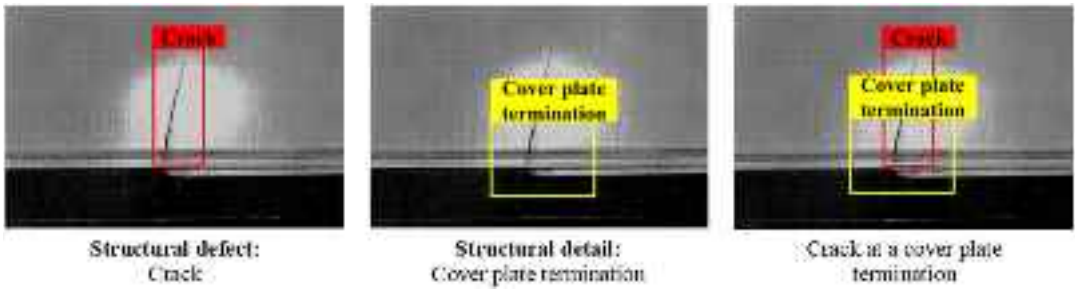

Fig. 1. Structural detail detection. [Reprinted from Haghani et al. (2012), under Creative Commons-BY-NC-SA-3.0 license (https://creativecommons.org/licenses/by-nc-sa/3.0/).]

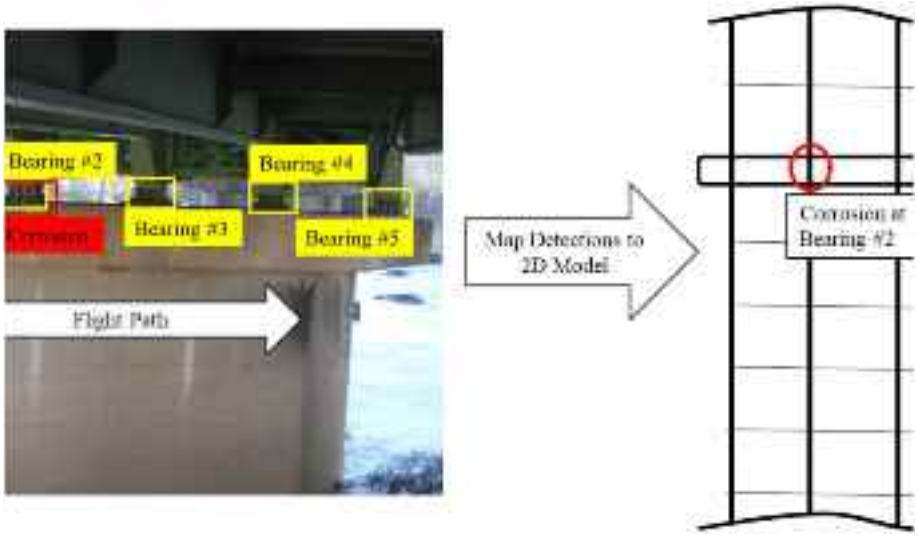

Fig. 2. Mapping detections to 2D model. (Image courtesy of Clark Nexen.)

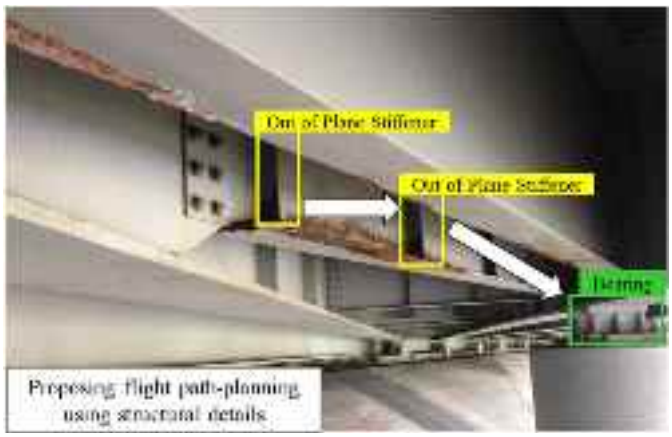

Fig. 3. UAV flight path planning using structural details. (Base image courtesy of Clark Nexsen.)

details (Fig. 1) (Haghani et al. 2012). Over the course of an UAV's flight, the computer vision may detect structural details like bearings and structural defects like corrosion. Bearings are at known locations on a bridge and so provide reference points for an UAV. Using structural detail reference points, any detected defect found near them could be contextualized to the structural detail reference. For example, given an UAV flight path with a known order and type of structural detail, the UAV could localize defect detections back onto a known two-dimensional (2D) model with the corresponding structural details (Fig. 2). During UAV flight, the detected structural details provide key objects in an UAV's vision to optimize the UAV flight path or to avoid crashing into a bridge while completing its inspection (Fig. 3).

Machine Learning Applications in Civil Infrastructure Inspection

Machine learning and deep learning techniques have seen increased applications in civil engineering, especially in the inspection process for defect detection (Cha et al. 2017; Delay et al. 2017; Dorafshan et al. 2016; Ellenberg et al. 2014; Feng et al. 2017; Rau et al. 2016; Sila Gulgec et al. 2019; Zhang et al. 2018; Zhou et al. 2016). Not all data sets or models have been publicly accessible. Because image-based machine learning for defect detection has become very popular, there is a need to continue making data and code publicly available for research on computer vision and artificial intelligence (AI) in inspections. It would be beneficial to the research community to consolidate the data sets and models of this similar domain into a data lake for more accessible consumption. This research makes both the model and the data used publicly findable, accessible, interoperable, and usable.

There are three levels of defect detection. The first is *identifying* that there is a defect, the second is *quantification*, and the third is the detecting of a *change* in defect over a specified timeframe.

Several papers have explored *classification* and object *identification* of defects, like machine learning to do edge detection for finding cracks (Guldur and Hajjar 2016; Prasanna et al. 2016). More recent research has used convolutional neural networks (CNNs) to identify issues like cracks in concrete (Cha et al. 2017) or pavement (Wang et al. 2017), and a CNN trained by the University of Illinois detected and identified multiple issues like spalling, corrosion, and cracking (Hoskere et al. 2018). *Quantifying* the extent of defects has been theorized using three-dimensional (3D) point clouds to obtain measurements from deconstructing the pixelwise measurements into actual measurements (Zheng et al. 2014; Mohan and Poobal 2017; Rau et al. 2016). The *change* in detection is far less common, but one study used a CNN model to detect changes in a tunnel over time (Stent et al. 2015). Another study detected change on a 3D point cloud (Shen et al. 2017). Much of the literature has explored classifying or quantifying defects, and most researchers have focused on crack detection (Hüthwohl et al. 2016).

Some papers similar to this research effort focus mostly on defect identification and classification. A CNN model, You Only Look Once (YOLO) version 3, for the object identification of defects was developed by researchers in Hong Kong (Zhang et al. 2018). It identified defect regions of interest using bounding boxes. One of the previously mentioned papers, from the University of Illinois, used semantic segmentation for its classification and identification of defects (Hoskere et al. 2018). In another paper, the authors created a data set that focused on the creation of a benchmark data set for concrete crack detection, SDNET2018 (Dorafshan et al. 2018). The most similar paper found had to do with detecting regions of interest that are susceptible to defects (Yeum et al. 2019). In Yeum's paper, regions of interest were bounded with boxes, but it only examined one type of connection; it was not concerned with bridge inspection, and it fell short in defining the type of connection. While the concept is similar, COCO-Bridge sets itself apart because it identifies multiple structural details, it was built for bridge inspection, and it classifies its predictions. Because of the recent success of CNNs in computer vision applications in civil and computer engineering that a CNN object detection model was chosen to detect structural bridge details in a scene.

Convolutional Neural Networks

CNNs were inspired by the visual cortex and have yielded groundbreaking results on image processing (Krizhevsky et al. 2012). CNNs are a part of what is known as deep learning. The term *deep* refers to the number of hidden layers in a neural network; the more layers, the deeper the network. For CNNs to learn or become intelligent, they must be trained. Image-processing CNNs are trained with a form of annotated image data corresponding to output classes.

There are four main categories in image-processing object detection that a CNN can produce: image classification, object localization, semantic segmentation, and masking. The four outputs require varying levels of effort to prepare and train. Each of these network types can be optimized and improved by adjusting the model architecture and feature learning techniques, as well as expanding or improving its training data set. There are several popular CNN architectures, including YOLO (Redmon et al. 2015), Single Shot Detector (SSD) (Liu et al. 2016), region-based CNN (R-CNN) (Ren et al. 2016), and GoogLeNet (Szegedy et al. 2014). Each has its own trade-offs in performance, speed, and training time.

The way that these models are compared against each other is through benchmark data sets. A benchmark data set is a data set that can be used to evaluate its model performance against a specific type of detection. For example, the Mixed National Institute of Standards and Technology (MNIST) data set is a handwritten number data set (LeCun et al. 1998). It consists of 60,000 images (32×32 pixels) of handwritten numbers and 10,000 test images. The International Conference on Document Analysis and Recognition (ICDAR) is a conference that regularly poses challenges on select data sets as part of an automated reading competition (Ibrahim et al. 2016). Object detection benchmarks includes the PASCAL Visual Object Classes challenge (Everingham et al. 2010) or ImageNet (Krizhevsky et al. 2012), the latter of which is an object-based recognition challenge. In all these challenges, the number of data points or images that operators can use to train their model is a controlling factor. Another controlling factor is the image pixel density because images must be rescaled to meet graphics processing unit (GPU) memory constraints. An alternative to manage data constraints is to expand the data set through image augmentation. Data augmentation has shown positive performance improvements (Chordia and Verma 2015).

Methodology

Overview

Four structural bridge details were chosen from the photos collected for this study: (1) bearings, (2) cover plate terminations, (3) gusset plates, and (4) out-of-plane stiffeners. These details were chosen because all of these structural details are damage-prone regions where bridge inspectors should typically focus increased effort owing to their potential for damage, and many readily available images contained these details. Quantity of data is typically the limiting factor when building a neural network. In this case, each structural bridge detail in each photo collected represented a data point.

Before collection of the photos, it was first necessary to define which types of structural detail classes would be targeted for training the neural network model. Unlabeled structural bridge detail images were outsourced from inspection firms, as were self-collected photos of specific bridge details from decommissioned plate girder bridges. Once the data were collected, each of the selected bridge details was annotated using bounding boxes to localize each region of interest. The images were split into a training set and an evaluation set. The images in the evaluation set were selected randomly from all the annotated data. Preprocessing of the data was done through project-specific Python version 3.6 scripts and using the TensorFlow version 1.10.0 application programming interface (API) (TensorFlow was developed by Google and may be used for machine learning and deep learning applications). The photos were rescaled to smaller dimensions using a project-specific Python script method, so that the V100 NVIDIA graphics processing units (GPUs) would not run out of memory during training (it was found that images over 600×600 pixels caused out of memory errors with the V100 NVIDIA GPUs). After rescaling, the images had the option to be augmented by mirroring or rotating or left as the originals. At this point, the images and their corresponding bounding box detail files (CSV files) were converted into TensorFlow records (tf.records). These special records, along with the training images, are what were referenced during the model training process. All the preprocessing was done with the developed Python code, which is referenced in the "Data Availability" section of the paper. With the tf.records, the training could begin. Pretrained CNN models can be found through open-source repositories. The pretrained network used in this project was the SSD network. Specifically, SSD version 2.0 (SSD v2) with pretrained neural network weights, from the COCO data set (Lin et al. 2016), was the CNN model used for training. Using pretrained

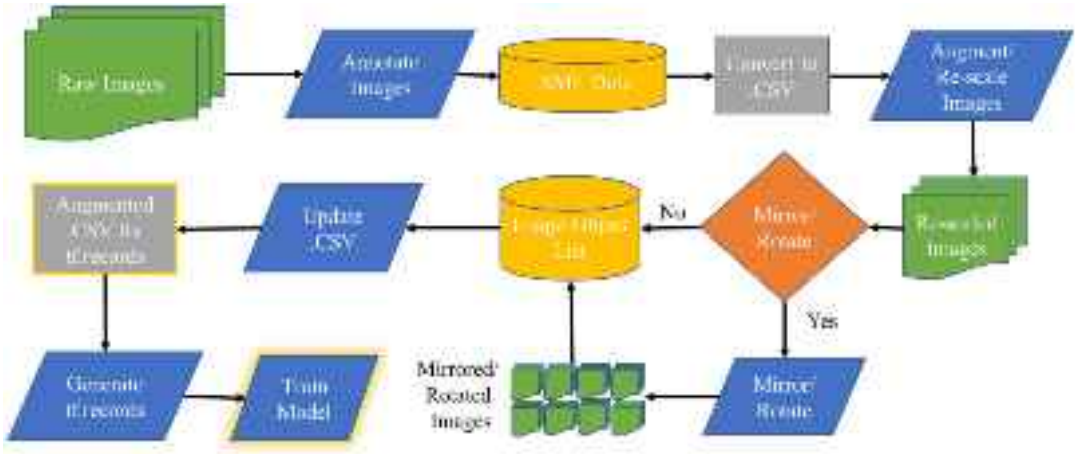

Fig. 4. Visual representation of data set construction.

weights as a starting point for training a neural network can help to reduce training time. Fig. 4 is a visual representation of the data set construction.

Chosen Structural Details—Data Set Classes

Table 1 summarizes the structural bridge details and their associated defects and fatigue-prone evaluation performed by an inspector (Ryan et al. 2012). Each defect has associated condition states with descriptions to guide an inspector to a decision on its rating. Condition states are on a scale of 1 (good) to 4 (severe). For example, if a bearing showed freckled rust, then the condition state would be rated by the inspector as 2 (fair) under the *corrosion* defect category.

Bearings transfer loads and movements of a bridge deck to the substructure and are inspected for corrosion, fatigue cracking, and rocking tolerance (ODOT 1973). Fig. 5 depicts three steel mechanical bearings. *Cover plate terminations* are a fatigue-prone detail that inspectors commonly review for fatigue cracks. Cover plates were once commonly used to improve the economy of bridges by reducing the overall amount of steel needed in a bridge section. They were popular from 1940 to 1970, but a few modern-day designs use them. The cover plate steel provides additional thickness to the flange section, which increases the section's flexural resistance. Cover plates may not extend the entire length of beams since they are used in regions of high flexural demand. It was found that these regions are prone to fatigue cracking, which typically initiates at the termination of the cover plate. Fig. 6 contains two pictures of cover plate terminations from the data collected for this study.

Gusset plate connections are common structural details found in many types of bridges, like trusses, suspension bridges, and girders. Gusset plates act as connection points at nodes where axial members meet and are susceptible to cracking, corrosion, section loss, and buckling, all of which reduce the effectiveness of the original design. An example of gusset plate failure would be the I-90 bridge over the Grand River in Ohio, when the bridge had to be shut down because a gusset plate had fractured (Bagnard 2016). Fig. 7 is an example of a gusset plate connection in a railroad box girder bridge. The fourth structural detail considered was the *out-of-plane stiffener* (Fig. 8). Out-of-plane stiffeners are welded or riveted onto a web or flange, and they increase the postbuckling shear capacity of a slender web section. Specific connection details of out-of-plane stiffeners have been known to be common fatigue-prone regions

Table 1. Structural bridge detail and associate defects

Chosen structural bridge details	Associated defects (FHWA)
Bearing	Corrosion, connection, movement, alignment, bulging, splitting, tearing, loss of bearing area
Cover plate termination	Fatigue-prone detail (cracking), section loss
Out-of-plane stiffener	Fatigue-prone detail (cracking), section loss
Gusset plate connection	Corrosion, cracking, connection, distortion, damage

Fig. 5. Sample bearings from collected images. (Images courtesy of Clark Nexsen.)

Fig. 6. Sample cover plate termination from collected data. (Images courtesy of Clark Nexsen.)

Fig. 7. Sample gusset plate connection from collected data. (Image by Eric Bianchi.)

Fig. 8. Sample out-of-plane stiffener from collected data. (Images by Eric Bianchi.)

(USDOT 2012). In particular, they have been prone to develop fatigue cracking from welded areas.

Preprocessing—Annotation and Augmentation

Following image data collection, the images were annotated using bounding boxes with their corresponding associated structural detail label. Bounding boxes are simple annotations that box regions of interest. Each of the boxes has a specific label corresponding to the object class that the box is encapsulating. These labeled boxes are what the neural network uses to train itself to locate and correctly identify objects. The annotation tool chosen in this paper was the open-source software LabelImg (Tzutalin 2015). This software can make annotations using bounding boxes and masks. Object detection using bounding boxes was chosen for this paper for its simplicity and speed. Whether the data are simple classifications, object detection, or mask R-CNN, researchers must have a tool for annotating the raw image data. The resulting output annotation files from LabelImg were in XML format, a universal mark-up language that can be easily converted into many desired formats.

The desired level of quality control was achieved through manual efforts using an annotator with a structural engineering background as opposed to an annotator who did not have the requisite technical background. Annotation of the images was not outsourced to third-party annotators like Amazon Mechanical Turk, as was done on some major data sets (Lin et al. 2016). Once the data were annotated, the images were rescaled to fit the GPU memory constraints and augmented depending on the test variation (Fig. 1). Augmenting and rescaling after annotation allowed the original data to remain as reference points. Images were rescaled from dimensions such as $5{,}184 \times 3{,}456$ pixels to dimensions no greater than 500×500 pixels. Although there exist many more ways to augment an image, only 90° incremental rotations and mirroring of the images about the y-axis were considered for augmentation. This made eight possible ways to augment the images. Take Fig. 9 as an example: one normal image can be augmented eight unique times. The authors of this paper consider this combination of mirroring and rotating to be *complete augmentation*.

The model was trained on the SSD v2 model with pretrained weights from the original COCO data set (Lin et al. 2016). The constant in the study was the model. This model was chosen because of its convenience, processing speed, and performance. Speed was important because the shorter the delay, the quicker the operator or the UAV can make a decision. Performance was important because the model must capture the structural details to initiate the decision-making process. The model configuration was not altered from the default configuration values. There were options to adjust the computational steps, batch size, learning rates, and other test executable variables, but only the number of computational steps was varied among tests. The test variations were compared using 5,000 computational steps for training the model. It is likely that other neural network models or model configurations exist that could outperform the SSD v2 model used in this paper, but the research efforts did not focus on optimizing a model for the data set.

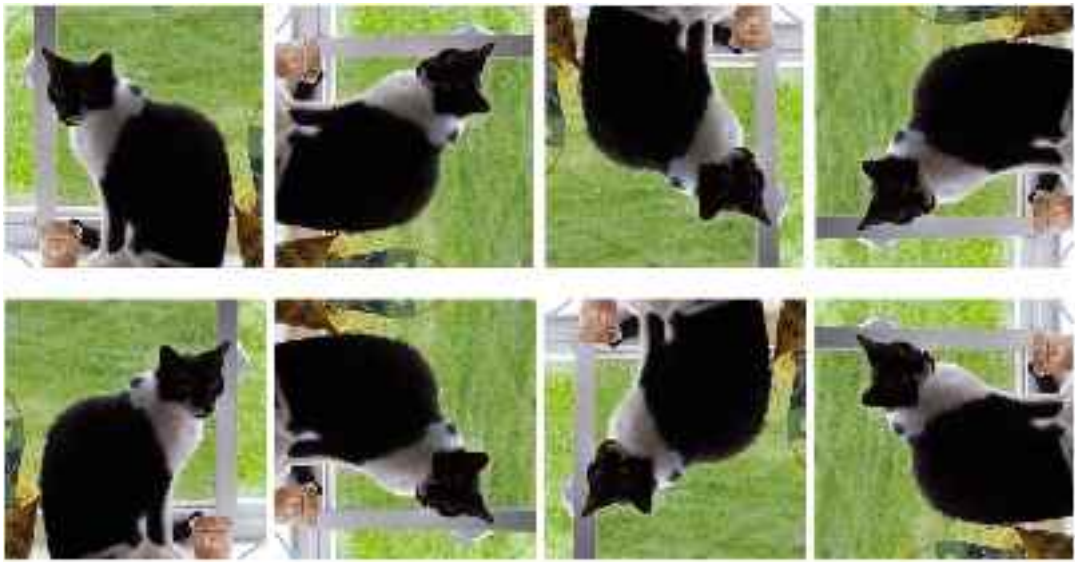

Fig. 9. Complete augmentation of an image. (Base image by Eric Bianchi.)

Table 2. Data set composition

Model	Gusset plate instances	Out-of-plane stiffener instances	Cover plate termination instances	Bearing instances	Total number of unique images	Image scaling
Complete augmentation	4,504	8,816	1,576	3,800	5,752	(256–400) × (230–400)
Selective augmentation	1,126	2,204	394	950	1,438	(256–400) × (230–400)
Control	563	1,102	197	475	719	(256–400) × (230–400)
Evaluation data set ratio to training data set	89 (15.8%)	116 (10.5%)	14 (7.1%)	27 (5.7%)	55	(256–400) × (230–400)

Test Variations

This data set included 764 images with 2,583 structural detail instances. There are many more structural detail instances than images because a single image often had multiple structural details within it. The validation set was split by a 1:10 ratio of all the data. A single image may have multiple structural details within it. The control model was trained using all the images with no image augmentation. Two main validation variations were evaluated against the control model: *complete augmentation* and *selective augmentation*. The model variations were evaluated with the same evaluation data set. All model configurations, like the learning rate, batch size, and gradient descent optimizer, were kept the same. Table 2 summarizes the training and evaluation characteristics of each model. The motivation for *completely augmenting* an image (as shown in Fig. 10) was to increase the number of unique images of the data set, thereby increasing the model's exposure to the structural detail. The theory was that more images and object instances would yield a stronger prediction model. Additionally, the augmentation variations would theoretically introduce randomness to help build a more robust neural network.

When the training of the *complete augmentation* was complete, the results were compared to the *control model*. The model accuracy and precision decreased; *complete augmentation* hurt the model learning. It was hypothesized that this was because the augmentation introduced structural detail orientations that realistically are never present, such as a bridge bearing oriented 90° from the horizontal plane. It was theorized that mirroring the original image along only the y-axis maintained the geometry fidelity and realistic nature of the structural detail, as shown in Fig. 11. This type of image augmentation was considered *selective augmentation*.

Postprocessing—Evaluating Model Performance

Typical postprocessing techniques for object detection were used to evaluate model performance. Mean average precision (mAP) and F1 scores were used when evaluating the model performance in accordance with the literature (Girshick et al. 2014; Hu et al. 2017; Ren et al. 2016; Siam et al. 2017). Typically in the literature, mAP scores are used to evaluate models, but mAP is not as practical in this design because accuracy was not considered. Therefore, F1 scores were chosen as another method to evaluate model performance. F1 scores consider both accuracy and precision with equal weighting. Whether F1 scores or mAP is used, *evaluation thresholds* must be set. Adjustments to these thresholds affect the scores of these performance metrics.

This paper defines *evaluation thresholds* as the prediction confidence threshold (λ) and the intersection over union (IOU) threshold. The confidence threshold for a model determines whether to consider a prediction as valid. The λ indicates how confident the model is that it recognizes an object it has been trained to look for in an image. If the model is at or above λ, then the model will record the prediction; otherwise, it will suppress the prediction. This is an important parameter since an inspector or an UAV would not want to muddle its view with low-confidence, incorrect predictions. IOU is another object detection evaluation parameter that may be adjusted at the discretion of the operator. The IOU measures how well a model predicts the location and extent of an object in images.

Fig. 10. Complete augmentation. (Base images courtesy of Clark Nexsen.)

Fig. 11. Selective augmentation. (Base images courtesy of Clark Nexsen.)

The IOU is the threshold that the prediction, i.e., bounding box, matched the ground truth bounding box. The IOU is calculated by taking the area of the predicted bounding box and dividing it by the union of the ground truth bounding box and the predicted bounding box. Ground truth bounding boxes are drawn around objects prior to training and are used to help train and evaluate the model. The better the match of the model prediction to the ground truth, the higher the IOU percentage. *Evaluation thresholds* that are too high may suppress correct outputs. With *evaluation thresholds* set too low, many false positives may be introduced. In the literature, it is common to compare model performance using an IOU of 50% for the mAP score (Girshick 2015; He et al. 2017; Veit et al. 2016), and in some cases 75% (Hu et al. 2017).

Results

The confidence level (λ) from the model evaluation were outputted quantitatively, as percentages, and qualitatively. The qualitative results showed model predictions and ground truth bounding boxes superimposed over images. These qualitative results may be found in the appendixes, showing the evaluation data set and a sample output of predictions of the control model. IOU and confidence thresholds are the two main postprocessing parameters to use when evaluating an object detection model. A confidence level that is too low will crowd the output image with many erroneous false predictions, as shown in Fig. 12(a). A confidence level that is too high may suppress many correct predictions, as shown in Fig. 12(c), where the middle gusset plate was not accurately identified. Fig. 12(b) is an optimal confidence threshold, correctly predicting all structural details, without any false predictions.

The confidence thresholds are a crucial setting that allows operators to refine model output. Four confidence thresholds were chosen to evaluate model performance: 10%, 25%, 50%, and 75%, where 10% was chosen as the lower bound because confidence thresholds lower than that gave unreasonable and impractical predictions [Fig. 12(a)]. These confidence thresholds were used when the models were evaluated by mAP and F1 scores.

Two IOU thresholds were used to evaluate the images: 35% and 50%. Because some of the details have difficult geometry, the model had a hard time achieving an IOU of 50%. For example, the out-of-plane stiffeners are long and narrow (Fig. 13), and the cover plate terminations are small relative to the overall image size (Fig. 14). The models showed significant decreases in precision

J. Comput. Civ. Eng., 2021, 35(3): 04021003

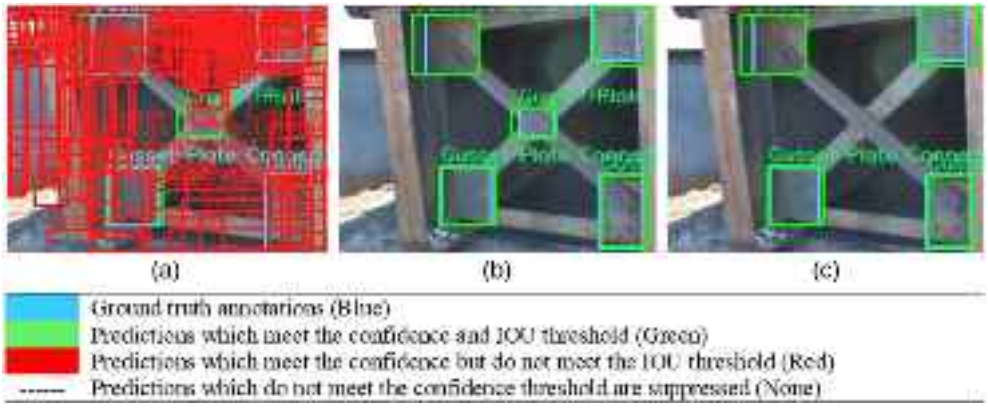

Fig. 12. Confidence thresholds from control model where (a) $\lambda = 1\%$; (b) $\lambda = 10\%$; and (c) $\lambda = 75\%$. (Base image by Eric Bianchi.)

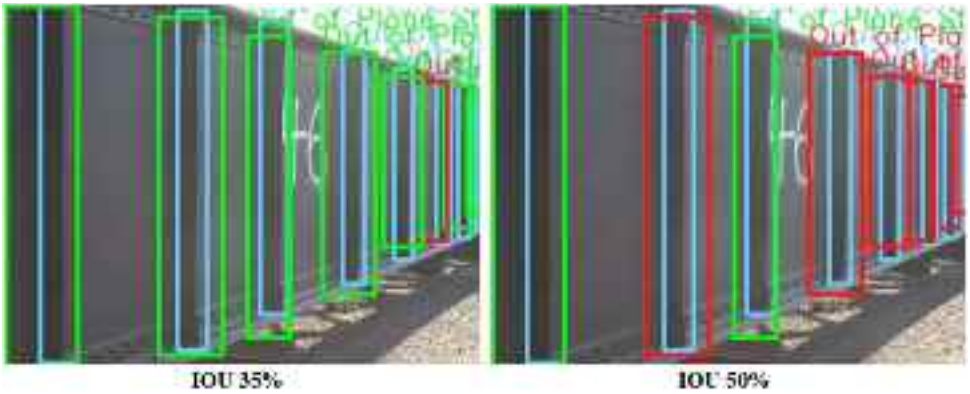

Fig. 13. Narrow evaluation criteria for out-of-plane stiffeners from control model. (Base image by Eric Bianchi.)

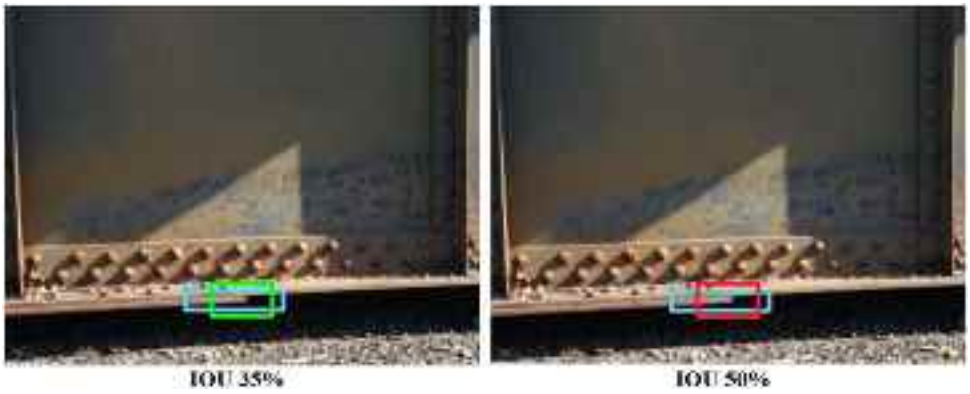

Fig. 14. Small evaluation criteria for cover plate termination from control model. (Base image by Eric Bianchi.)

when evaluated at the two different IOUs of 35% and 50%. The out-of-plane stiffeners and cover plate terminations decreased in precision by 26.5% and 25%, respectively, while the bearings and gusset plate connections decreased by less than 10% (Table 3). Given the choice between an IOU of 35% and an IOU of 50%, the 35% IOU threshold would be the more reasonable perspective for the model evaluation.

The following equation captures the overall decline in accuracy and precision of the model from the IOU threshold:

$$\frac{\sum_{\text{Model 1}}^{\text{Model 3}} \sum_{10\%}^{75\%} \frac{(\text{IOU}_{35}\lambda_i) - (\text{IOU}_{50}\lambda_i)}{(\text{IOU}_{35}\lambda_i)}}{12}$$

where there are three models (*control*, *selective augmentation*, *complete augmentation*), and each model captured the percentage drop in accuracy/mAP from the subsequent IOU at each confidence threshold ($\lambda = 10\%$, 25%, 50%, 75%). The summation of the confidence thresholds was added to each model and divided by 12 (3 models × 4 confidence thresholds). This was effectively the average percentage loss of performance between IOU thresholds.

The results of the three models (*control*, *selective augmentation*, *complete augmentation*) were evaluated using mAP and F1 scores. Table 4 summarizes a portion of the results from the confidence threshold of 25%. It was clear from these results that *complete augmentation* did not perform as well as *selective augmentation*

or the *control*. Therefore, Table 5 only compares the difference in performance between the *selective augmentation* and the *control* for each evaluation method (i.e., AP^{35} and λ^{25} $45.0\% - 44.1\% = 0.9\%$). A positive value indicates that selective augmentation performed better than the control, whereas a negative value indicates that the control performed better than selective augmentation.

Table 3. Average percentage loss of performance between IOU thresholds

Evaluation	IOU (%)	Bearing (%)	Cover plate termination (%)	Gusset plate connection (%)	Out-of-plane stiffener (%)
mAP	35–50	9.62	26.55	5.06	25.00
Accuracy	35–50	8.47	14.73	4.33	14.92

Table 4. Model performance comparisons with a confidence threshold of 25%

Performance evaluation method	IOU %	Control	Selective augmentation	Complete augmentation
mAP	AP^{35}	44.1%	45.0%	34.2%
	AP^{50}	33.9%	36.4%	28%
F1 score	AP^{35}	0.594	0.618	0.512
	AP^{50}	0.506	0.528	0.431

Table 5. Performance of selective augmentation versus control

Performance evaluation method	IOU %	λ^{10}	λ^{25}	λ^{50}	λ^{75}
mAP	AP^{35}	(−5.1%)	0.9%	5.4%	5.0%
	AP^{50}	2.6%	2.5%	5.3%	4.9%
F1 scores	AP^{35}	(−0.016)	0.024	0.063	0.082
	AP^{50}	0.037	0.024	0.073	0.085

Note: Values in parenthesis helps to better-distinguish negative values.

Fig. 15. Missed gusset plate connection (red), distant bearings (yellow). (Base image courtesy of Clark Nexsen.)

Future Improvements

The main avenues for improvement relate to how the data set is annotated, how the data set is split between training and testing, and possibly from how the models are trained. There were no established annotation guidelines to follow when annotating the images for structural details. Creating consistent guidelines on how to annotate the data would help ensure that a data set was consistent and the model outputs predictable. For example, Fig. 15 shows some distant bearings annotated with a yellow bounding box. These bearings are far away in the picture, so they were not considered in the annotation. However, the annotation distance was never defined. These types of annotation decisions can affect model performance. In the future, consistent annotation quality control checks will help to keep the output closer to the expected result. In Fig. 16, the annotation, which boxed the out-of-plane stiffener, should have been drawn more precisely. The image shows a yellow dashed line, which is what was drawn in the data set, and a red solid line, which is the ideal bounding box. Additionally, some structural details in the images were ignored since only slivers of them were shown in the picture. However, after viewing results from some initial trials, it was observed that the model was able to correctly predict these slivers of structural details. Thus, the data set was adjusted to encapsulate these correct predictions by including more annotations around partially obstructed structural details. Still, it was noted that guidelines for the annotation of such partial details were necessary. Without standard annotation rules, the model output is less predictable and more inconsistent. Furthermore, some structural details were simply missed during the annotation process, like the gusset plate annotated with a red bounding box in Fig. 15. Missing structural details confuses the model by leaving out a positive structural detail instance that the model may now learn to ignore.

As more structural detail data are acquired, the data set split will be adjusted to be more consistent across each of the classes. For example, the *gusset plate connections* had 15.77% object instances in the evaluation data set, while *bearings* had only 4.7% (Table 2). Maintaining a more equal ratio across all classes allows for more reliable model evaluations.

A more robust model may use more sophisticated methods of training and achieve a stronger AI model. Some of the training parameters, such as batch size, were arbitrarily chosen to be a reasonable number (24), and other parameters, such as the learning rate, were not adjusted. Part of the reasoning behind this decision was so as not to introduce more independent variables, which

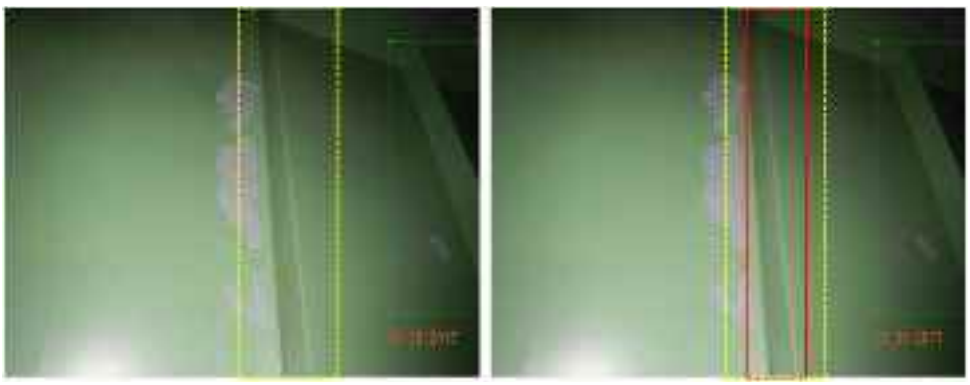

Fig. 16. Bounding box inconsistencies. (Base images courtesy of Clark Nexsen.)

would increase the scope of the project. Because these test configuration parameters were not tuned, and only one model was used, there is room for growth and exploration for developing a tuned model for structural detail detection.

Discussion

The findings of this study clearly indicate that the selective augmentation model was stronger than the control model for every confidence threshold and IOU (except $\lambda^{10}AP^{35}$ for both F1 scores and mAP). Thus, selective augmentation is a promising method of preprocessing data to enhance model performance. The complete augmentation of the data consistently underperformed both the control and the selective augmentation. Thus, complete augmentation is not suggested for bridge inspection data or other images requiring context. Across all models (*control, selective augmentation*, *complete augmentation*), there was significant loss in mAP and accuracy for *cover plate terminations* and *out-of-plane stiffeners* between IOUs 35% and 50%. This finding suggests that, owing to the geometry and possibly the way the structural details were annotated, the model had a more difficult time interpreting the desired ground truth bounding box. *Bearings* and *gusset plate connections* are generally square in shape, whereas *out-of-plane stiffeners* and *cover plate terminations* are more rectangular. Long and

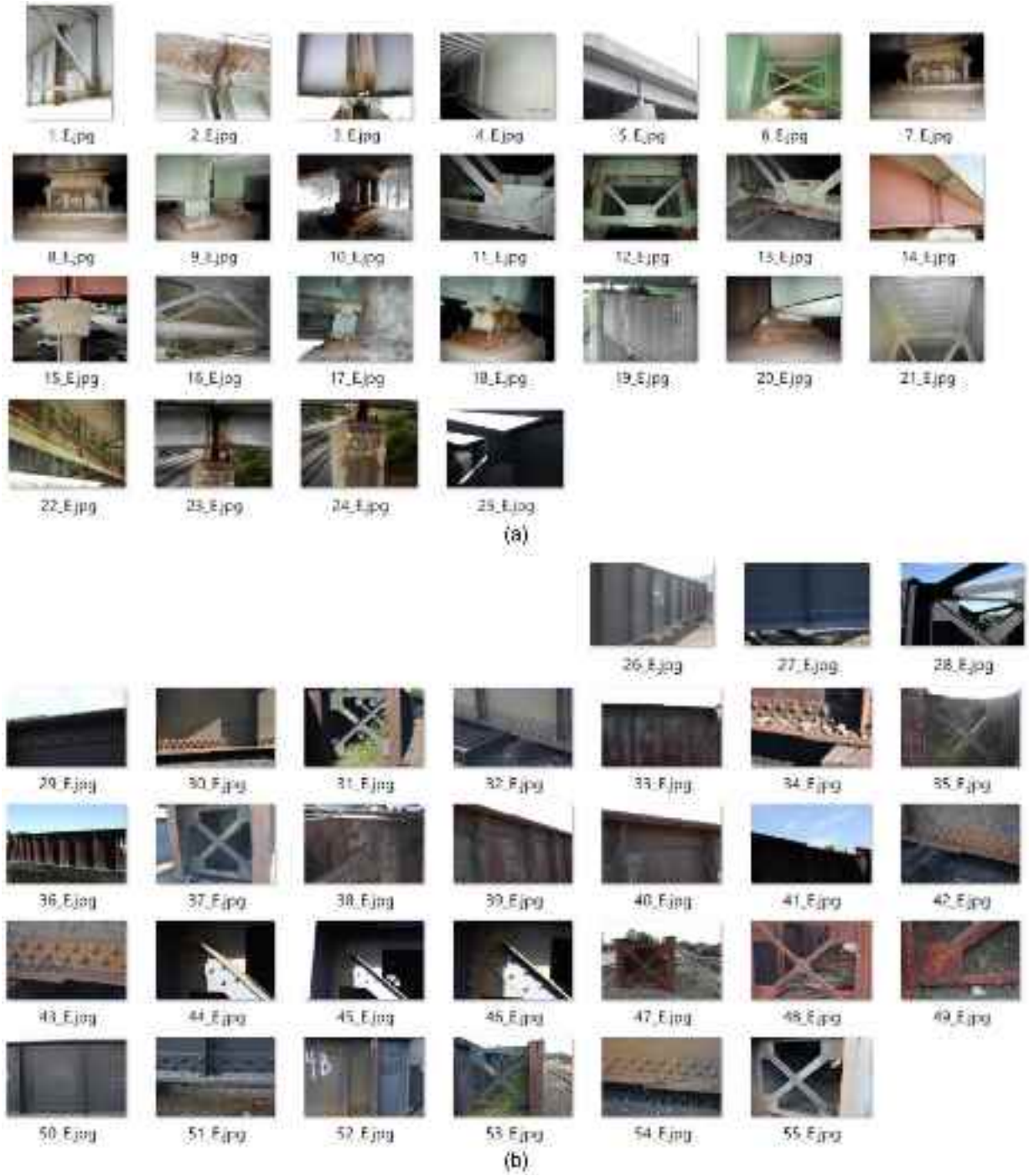

Fig. 17. Evaluation data set images. [Images in (a) courtesy of Clark Nexsen; base images in (b) by Eric Bianchi.]

narrow shapes had a tougher time adhering to stricter IOUs. When the model's prediction was slightly askew, it rendered massive percentage area differences. Additionally, defining a *cover plate termination* with a box was difficult because such regions are more of a single point than a specific area. Therefore, the ground truth boxes were ambiguous in nature as to where they start and stop. With such an unspecified method of annotating cover plate terminations and unestablished guidelines, the model had a difficult time fitting to the desired ground truth output, as seen in Fig. 14.

Conclusions

- It was found that in this domain of data, structural details, the context of the training images mattered in the model's performance. The training images that maintained the contextual environment (mirrored about the *y*-axis) performed better than the training images that did not (complete augmentation). It is suggested that the contextual environment be maintained when training data to extend the performance of a data set for a CNN model.
- It is recommended that before deploying a CNN model in the field for inspection, one should determine a confidence threshold that is functional for its purpose As shown in Fig. 9, it is important to test the confidence threshold sensitivity.
- Throughout this study and upon its completion, procedures are recommended for developing a structural detail data set and CNN model. Specifically, having clearly defined rules for annotation will help improve the quality of the data set. This paper only used one researcher to annotate the images. Introducing inter-rater reliability, or a second human to review the work of another, will add a layer of accuracy and precision. The second pass would be inherently quicker, focusing on the adherence to the guidelines and capturing all the details. These recommendations will minimize inconsistencies and missed structural details in future annotation cycles.

Fig. 18. Predictions for IOU of 35% and confidence threshold of 25% for the control model. [Base images in (a) by Clark Nexsen; base images in (b) by Eric Bianchi.]

- To follow the standards in CNN model evaluations found in the literature, it is recommended that data be split into a 15%–20% evaluation set and 80%–85% training images.
- Details should be evaluated in a balanced manner, and the fraction of object instances should be consistent across all classes. For example, if there were 2,000 object instances of out-of-plane stiffeners and 1,000 gusset plate connections and the data were split into 20% evaluation, 80% training, then there should be approximately 400 object instances of out-of-plane stiffeners and 200 gusset plate connections in the evaluation data set to ensure that the data set is equally balanced among different classes.

Appendix I. Evaluation Data Set

Fig. 17 shows the evaluation data set images.

Appendix II. Sample Output of Predictions of Control Model

Fig. 18 shows the predictions for an IOU of 35% and confidence threshold of 25% for the control model.

Data Availability Statement

Some or all data, models, or code generated or used during the study are available in a repository or online in accordance with funder data retention policies (https://doi.org/10.7294/M8PG-4A02). Additionally, extensions of these data and more are available at www.coco-bridge.com.

Acknowledgments

The authors would like to thank the National Science Foundation and the Center for Unmanned Aircraft Systems. The authors would like to also thank the following companies for their donation of time and data: Norfolk Southern, Roanoke, Clark Nexsen, Roanoke, AECOM, Roanoke, Michael Baker International, and the Virginia DOT.

Notation

The following symbols are used in this paper:

AP^{35} = intersection over union of 35%;
AP^{50} = intersection over union of 50%;
λ^{10} = confidence threshold of 10%;
λ^{25} = confidence threshold of 25%;
λ^{50} = confidence threshold of 50%; and
λ^{75} = confidence threshold of 75%.

References

Bagnard, M. 2016. "Gusset plate inspection issues." Accessed November 22, 2018. https://www.ntsb.gov/news/events/Documents/minneapolis_mn-9_Gusset_Plate_Inspection_Issues.pdf.

Cha, Y.-J., W. Choi, and O. Büyüköztürk. 2017. "Deep learning-based crack damage detection using convolutional neural networks." *Comput.-Aided Civ. Infrastruct. Eng.* 32 (5): 361–378. https://doi.org/10.1111/mice.12263.

Chordia, S., and R. Verma. 2015. "Plankton image classification." Accessed July 23, 2018. http://cs231n.stanford.edu/reports/2015/pdfs/sagarc14_final_report.pdf.

Delay, B. L., M. Moaveni, J. M. Hart, P. Sharpe, and E. Tutumluer. 2017. "Evaluating railroad ballast degradation trends using machine vision and machine learning techniques." Accessed April 16, 2019. https://ascelibrary-org.ezproxy.lib.vt.edu/doi/pdf/10.1061/9780784480441.045.

Dorafshan, S., M. Maguire, and X. Qi. 2016. "Automatic surface crack detection in concrete structures using OTSU thresholding and morphological operations." Accessed June 9, 2019. https://digitalcommons.usu.edu/cee_facpub.

Dorafshan, S., R. J. Thomas, and M. Maguire. 2018. "SDNET2018: An annotated image dataset for non-contact concrete crack detection using deep convolutional neural networks." *Data Brief* 21: 1664–1668. https://doi.org/10.1016/j.dib.2018.11.015.

Ellenberg, A., A. Kontsos, I. Bartoli, and A. Pradhan. 2014. "Masonry crack detection application of an unmanned aerial vehicle." Accessed April 16, 2019. https://ascelibrary-org.ezproxy.lib.vt.edu/doi/pdf/10.1061/9780784413616.222.

Everingham, M., L. Van Gool, C. K. I. Williams, and J. Winn. 2010. "The PASCAL visual object classes (VOC) challenge." *Int. J.* 303–338. https://doi.org/10.1007/s11263-009-0275-4.

Feng, C., M. Y. Liu, C. C. Kao, and T. Y. Lee. 2017. "Deep active learning for civil infrastructure defect detection and classification." In *Proc., Congress on Computing in Civil Engineering*, 298–306. Reston, VA: ASCE. https://doi.org/10.1061/9780784480823.036.

Gillins, D., C. Parrish, M. Gillins, and C. Simpson. 2018. "Eyes in the sky: Bridge inspections with unmanned aerial vehicles." Accessed June 9, 2018. https://www.oregon.gov/ODOT/Programs/ResearchDocuments/SPR787_Eyes_in_the_Sky.pdf.

Girshick, R. 2015. "Fast R-CNN." In *Proc., IEEE Int. Conf. on Computer Vision*. New York: IEEE. https://doi.org/10.1109/ICCV.2015.169.

Girshick, R., J. Donahue, T. Darrell, and J. Malik. 2014. "Rich feature hierarchies for accurate object detection and semantic segmentation." Accessed June 14, 2018. http://www.cs.berkeley.edu.

Guldur, B., and J. F. Hajjar. 2016. "Automated classification of detected surface damage from point clouds with supervised learning." Accessed June 8, 2018. http://www.iaarc.org/publications/fulltext/ISARC2016-Paper057.pdf.

Haghani, R., M. Al-Emrani, and M. Heshmati. 2012. "Fatigue-prone details in steel bridges." *Buildings* 2 (4): 456–476. https://doi.org/10.3390/buildings2040456.

Hallermann, N., G. Morgenthal, and V. Rodehorst. 2015. "Unmanned aerial systems (UAS)—Case studies of vision based monitoring of ageing structures." Accessed June 9, 2018. https://www.ndt.net/article/ndtce2015/papers/169_hallermann_norman.pdf.

He, K., G. Gkioxari, P. Dollar, and R. Girshick. 2017. "Mask R-CNN." In *Proc., IEEE Int. Conf. on Computer Vision*. New York: IEEE. https://doi.org/10.1109/ICCV.2017.322.

Hoskere, V., Y. Narazaki, T. Hoang, and B. Spencer. 2018. "Vision-based structural inspection using multiscale deep convolutional neural networks." Accessed May 26, 2018. http://arxiv.org/abs/1805.01055.

Hu, H., J. Gu, Z. Zhang, J. Dai, and Y. Wei. 2017. "Relation networks for object detection." Accessed November 19, 2018. http://openaccess.thecvf.com/content_cvpr_2018/papers/Hu_Relation_Networks_for_CVPR_2018_paper.pdf.

Hüthwohl, P., R. Lu, and I. Brilakis. 2016. "Challenges of bridge maintenance inspection." In *Proc., 16th Int. Conf. on Computing in Civil and Building Engineering*. Cambridge, UK: Univ. of Cambridge. https://doi.org/10.17863/CAM.26634.

Ibrahim, A., A. L. Abbott, and M. E. Hussein. 2016. "An image dataset of text patches in everyday scenes." Accessed June 10, 2018. https://arxiv.org/pdf/1610.06494.pdf.

Krizhevsky, A., I. Sutskever, and G. E. Hinton. 2012. "ImageNet classification with deep convolutional neural networks." Accessed June 9, 2018. https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

LeCun, Y., L. Bottou, Y. Bengio, and P. Haffner. 1998. "Gradient-based learning applied to document recognition." *Proc. IEEE* 86 (11): 2278–2323. https://doi.org/10.1109/5.726791.

Lin, T.-Y., M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. 2016. "Microsoft COCO: Common objects in context." Preprint, submitted May 1, 2014. http://arxiv.org/abs/1405.0312.pdf.

Liu, W., D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. 2016. "SSD: Single shot multibox detector." Accessed June 10, 2018. https://github.com/weiliu89/caffe/tree/ssd.

Mohan, A., and S. Poobal. 2017. "Crack detection using image processing: A critical review and analysis." *Alexandria Eng. J.* 57 (2): 787–798. https://doi.org/10.1016/J.AEJ.2017.01.020.

ODOT. 1973. "Ohio Department of Transportation manual of bridge inspection." Accessed November 20, 2018. http://www.dot.state.oh.us/Divisions/Engineering/Structures/bridge%20operations%20and%20maintenance/Manual%20of%20Bridge%20Inspection%202014%20v8/Manual%20of%20Bridge%20Inspection%202014%20v8%20without%20Appendix.pdf.

Otero, L. D. 2015. *Proof of concept for using unmanned aerial vehicles for high mast pole and bridge inspections*. Rep. No. BDV28-977-02. Tallahassee, FL: Florida DOT.

Prasanna, P., et al. 2016. "Automated crack detection on concrete bridges." *IEEE Trans. Autom. Sci. Eng.* 13 (2): 591–599. https://doi.org/10.1109/TASE.2014.2354314.

Rau, J. Y., K. W. Hsiao, J. P. Jhan, S. H. Wang, W. C. Fang, and J. L. Wang. 2016. "Bridge crack detection using multi-rotary UAV and object-base image analysis." Accessed June 10, 2018. http://uavg17.ipb.uni-bonn.de/wp-content/papercite-data/pdf/rau2017uavg-54.pdf.

Redmon, J., S. Divvala, R. Girshick, and A. Farhadi. 2015. "You only look once: Unified, real-time object detection." Accessed November 27, 2018. http://pjreddie.com/yolo/.

Ren, S., K. He, R. Girshick, and J. Sun. 2016. "Faster R-CNN: Towards real-time object detection with region proposal networks." Preprint, submitted June 4, 2015. http://arxiv.org/abs/1506.01497.pdf.

Ryan, T. W., J. E. Mann, and Z. M. Chill. 2012. "FHWA bridge inspector's reference manual (BIRM) (Vol. 1)." Accessed November 22, 2018. https://www.fhwa.dot.gov/bridge/nbis/pubs/nhi12049.pdf.

Shen, Y., R. Lindenbergh, and J. Wang. 2017. Change analysis in structural laser scanning point clouds: The baseline method." *Sensors (Switzerland)* 17 (1): 1–25. https://doi.org/10.3390/s17010026.

Siam, M., S. Elkerdawy, M. Jagersand, and S. Yogamani. 2017. "Deep semantic segmentation for automated driving: Taxonomy, roadmap and challenges." Preprint, submitted July 8, 2017. http://arxiv.org/abs/1707.02432.pdf.

Sila Gulgec, N., M. Takáč, and S. N. Pakzad. 2019. "Convolutional neural network approach for robust structural damage detection and localization." *J. Comput. Civ. Eng.* 33 (3): 04019005. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000820.

Stent, S., R. Gherardi, B. Stenger, and R. Cipolla. 2015. "Detecting change for multi-view, long-term surface inspection." Accessed June 8, 2018. http://mi.eng.cam.ac.uk/~cipolla/publications/inproceedings/2015-BMVC-change-detection.pdf.

Szegedy, C., W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. 2014. "Going deeper with convolutions." Accessed December 18, 2018. https://www.cs.unc.edu/~wliu/papers/GoogLeNet.pdf.

Tzutalin. 2015. "LabelImg. Git code." Accessed April 16, 2018. https://github.com/tzutalin/labelImg.

USDOT. 2012. "Steel bridge design handbook design for fatigue." Accessed November 22, 2018. https://www.fhwa.dot.gov/bridge/steel/pubs/if12052/volume12.pdf.

Veit, A., T. Matera, L. Neumann, J. Matas, and S. Belongie. 2016. "COCO-Text: Dataset and benchmark for text detection and recognition in natural images." Accessed June 9, 2018. https://vision.cornell.edu/se3/wp-content/uploads/2016/01/1601.07140v1.pdf.

Wang, K. C. P., T. Allen, A. Yang, J. Cheng, and G. Gary. 2017. "Deep learning system for automated cracking survey & its performance with pixel accuracy: CrackNet." Accessed June 8, 2018. http://www.rpug.org/download/2017/6-2-Kelvin Wang.pdf.

Yeum, C. M., J. Choi, and S. J. Dyke. 2019. "Automated region-of-interest localization and classification for vision-based visual assessment of civil infrastructure." *Struct. Health Monit.* 18 (3): 675–689. https://doi.org/10.1177/1475921718765419.

Zhang, C., C. C. Chang, M. Jamshidi, and H. Kong. 2018. "Bridge damage detection using a single-stage detector and field inspection images." Preprint, submitted December 19, 2018. http://arxiv.org/abs/1812/1812.10590.pdf.

Zheng, P., C. Moen, C. Roberts-Wollmann, and T. Cousins. 2014. "Crack detection and measurement utilizing image-based reconstruction." Accessed June 8, 2018. https://vtechworks.lib.vt.edu/bitstream/handle/10919/48963/crack_detection_and_measurement_utilizing_image_based_reconstruction.pdf?sequence=1.

Zhou, S., Y. Chen, D. Zhang, J. Xie, and Y. Zhou. 2016. "Classification of surface defects on steel sheet using convolutional neural networks." *Mater. Technol.* 51 (1): 123–131. https://doi.org/10.17222/mit.2015.335.