6-19-2020

Automatic Delamination Segmentation for Bridge Deck Based on Encoder-Decoder Deep Learning Through UAV-based Thermography

Chongsheng
cheng.chongsheng@huskers.unl.edu

Zhexiong Shang
szx0112@huskers.unl.edu

Zhigang Shen
University of Nebraska - Lincoln, shen@unl.edu

Automatic Delamination Segmentation for Bridge Deck Based on Encoder-Decoder Deep Learning Through UAV-based Thermography

Chongsheng Cheng[1], Zhexiong Shang[2], and Zhigang Shen[3]

[1] Durham School of Architectural Engineering and Construction, University of Nebraska-Lincoln, 122 NH, Lincoln, NE 68588; e-mail: cheng.chongsheng@huskers.unl.edu

[2] Durham School of Architectural Engineering and Construction, University of Nebraska-Lincoln, 122 NH, Lincoln, NE 68588; e-mail: szx0112@huskers.unl.edu

[3] Durham School of Architectural Engineering and Construction, University of Nebraska-Lincoln, 113 NH, Lincoln, NE 68588; e-mail: shen@unl.edu

Abstract:

Concrete deck delamination often demonstrates strong variations in size, shape, and temperature distribution under the influences of outdoor weather conditions. The strong variations create challenges for pure analytical solutions in infrared image segmentation of delaminated areas. The recently developed supervised deep learning approach demonstrated the potentials in achieving automatic segmentation of RGB images. However, its effectiveness in segmenting thermal images remains under-explored. The main challenge lies in the development of specific models and the generation of a large range of labeled infrared images for training. To address this challenge, a customized deep learning model based on encoder-decoder architecture is proposed to segment the delaminated areas in thermal images at the pixel level. Data augmentation strategies were implemented in creating the training data set to improve the performance of the proposed model. The deep learning generated model was deployed in a real-world project to further evaluate the model's applicability and robustness. The results of these experimental studies supported the effectiveness of the deep learning model in segmenting concrete delamination areas from infrared images. It also suggested that data augmentation is a helpful technique to address the small size issue of training samples. The field test with validation further demonstrated the generalizability of the proposed framework. Limitations of the proposed approach were also briefed at the end of the paper.

Key words: Concrete Delamination; Thermography; Nondestructive Evaluation; Deep Learning; Encoder-Decoder Architecture; Semantic Segmentation; UAV.

1 Introduction

The deterioration of aged bridges has been recognized as a serious problem for structural safety and serviceability [1]. The horizontal debonding in the subsurface of the deck, known as deck delamination, often indicates the corrosion-induced deterioration of the deck reinforcement [2]. Locating and profiling the extent of delamination thus is an essential task for conducting bridge in-situ condition evaluations from the perspective of structural health monitoring (SHM). Nondestructive detection techniques (NDT) such as Infrared thermography (IRT) have been widely adopted for shallow delamination detection [3–7]. The challenge of using IRT lies in data interpretation and automatic processing. Previous studies have attempted to analyze the thermal images using different methods such as threshold for temperature histogram [8], k-mean clustering for temperature density [9], and region-growth for temperature spatial relationship [10,11]. So far, these reported methods required engineering judgment for parameter selection or hand-crafted feature

tuning. Essentially, in profiling the delamination by assigning the class of regions in the survey area at a dense level could help the engineer to recognize the area and the boundary of delamination from the condition map. This expectation exhibits a close tie to the concept of semantic segmentation in computer vision.

Image semantic segmentation in computer vision aims to divide an image into non-overlapped meaningful regions to retrieve the high level information such as object class and geometry in a scene [12]. Thus, the semantic segmentation is often referred to as the fine-grained inference for dense classification at the image pixel or super-pixel level [13]. Existing methods could be put into two major categories: unsupervised and supervised frameworks, and comprehensive reviews could be found in the literature [12–14]. Often, unsupervised methods required the utilization of hand-crafted features such as intensity, color, and SIFT, and thus a specific development of the model for different fields is desired. As a result, the generalization capability was often compromised when transferring the model from discipline to discipline. Supervised methods provided a framework to train the model to learn the labeled data and generate learned features from data instead of handcrafting. It often achieved state-of-art performance in scene parsing tasks through better handling of contextual information [12].

For the task of delamination profiling through thermography, the challenge exists in the delamination feature generalization: 1) the shape and the depth of delamination varies, which leads to indeterministic feature geometry and contrast; 2) environmental factors such as air temperature and solar intensity varies during the day, which introduces the feature variation of the same delamination; 3) surface textures such as cracks, color difference, patching, and road painting adds external noise and the superimposition of non-favorable features. Thus, only using the hand-crafted features for delamination profiling required sophisticated designs of the model or fine-tuning of parameters to account for these case-by-case variations [8,10,15–17]. In comparison, the learned features from a supervised framework could address the issue when including these variations into the training data and thus increase the generalization ability of the model [18]. However, this framework demands: 1) certain levels of customizations to existing architectures of models for the specific task; 2) an adequate amount of annotated data with rich diversity for training to enrich the ability of the model.

Given the aforementioned challenges, this paper proposes a deep learning model using encoder-decoder architecture for automated delamination profiling. To demonstrate the capability of the proposed model, the experimental studies were conducted to collect the thermographic data of concrete slabs with artificial delamination embedded in different depths under natural environment. Then, the data augmentation strategies were implemented to enrich the sample variance for the model training. The model performance and the effect of the augmentation strategies are evaluated through multiple datasets. Finally, a procedure named, dense sliding window detector, was proposed to automatically process and compose the model predictions for the bridge implementation. The post-processing method by the conditional random field was used for additional refinement of final segmentation.

2 Background

2.1 The Characteristic of Delamination in Thermal Images

The pattern of the delamination shown in the thermal image is often recognized as the local abnormal region. Depending on the time of observation, it appears as a hot region during the heat-absorbing stage and a cold region during the heat-release stage. The mechanism behind the phenomenon is the change of thermal properties of the bridge deck due to horizontal cracking around the rebar level. This means the heat conduction rate is different under solar radiation during the day [19]. As a result, the pattern would be varied due to different sizes and depths of the acute delamination, as well as different time windows when the data was collected. **Fig.1** illustrates several variations observed by constructed samples under the natural thermal cycle of the day (see **Fig. 6** for the experimental design and environment). It shows the thermal images of three mimicked delaminations buried at different depths in the concrete slabs at different time windows. The red dash box indicates the real size of the delamination. **Fig.1a** shows the thermal image of the delamination at 11 am with a depth of 4.5 cm (left), 7 cm (middle), and 9.5 cm (right) from the top surface. The shallower delamination has more temperature contrast than the deeper one at the same time window. **Fig.1b**

shows the temperature profile of the section (black dash line) at a different time of the day. It reveals that the average temperature shifts at different time windows for the same slab. **Fig.1c** shows the preprocessed image by standardization (**Eq.1**), which is useful to remove the effect of temperature shifts.

$$I_s = \frac{I - m_I}{\delta_I}, \qquad (1)$$

Where I refers to the raw thermal image, m_I is the mean temperature of I, δ_I is the standard deviation of I. The standardized image I_s represents the temperature deviation from its own mean at the current time. Through the **Fig.1c**, the contrast variation between different depths and time windows are more clearly represented. Additionally, the standardization also centers and normalizes the data, which is beneficial to the model convergence during the training by the deep learning framework [20].

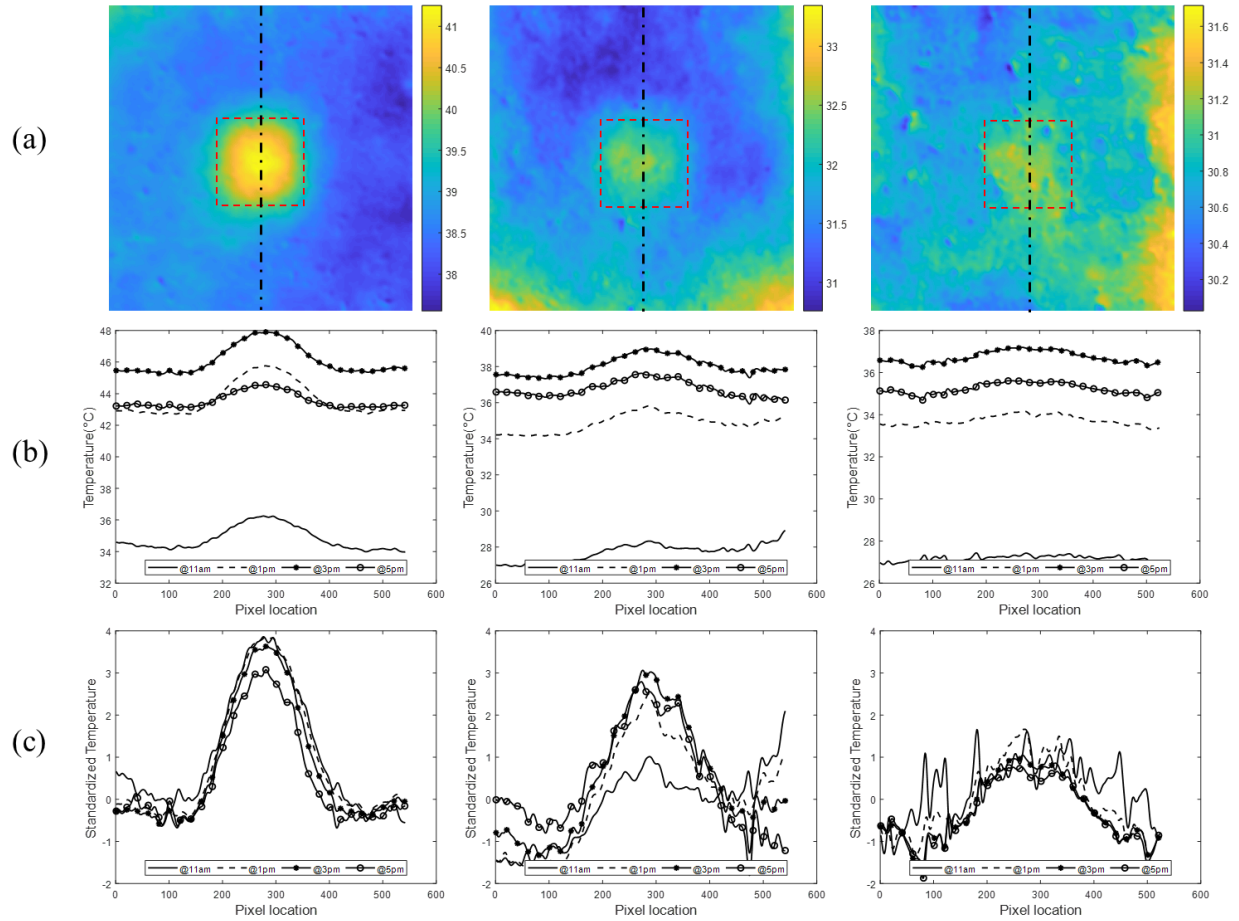

Figure 1. The pattern of simulated delamination in the thermal image at different time windows: (a) raw thermal image of the delamination at 11 am for 4.5 cm (1.75") deep (left), 7 cm (2.75") deep (middle), and 9 cm (3.75") deep (right); (b) section profile of the temperature at different time windows; (c) section profile after standardization of (b).

2.2 Encoder-Decoder Architecture for Semantic Segmentation

The encoder-decoder architecture has been widely used for pixel-wise classification for image semantic segmentation [13]. The concept behind this is the architecture built-up based on the convolutional neural network (CNN), which is designed to detect the local visual motifs of the object [21]. Generally, the image is first passed through the encoder part where the spatial information is convolved with the filter banks in each layer to produce a set of feature maps in a down-sampling manner. Thus, the information is encoded in dense representations with invariance to the spatial locations. The decoder part of the architecture translates the latent information in an up-sampling manner to the spatial locations to present segmentation results. The common architectures such as AlexNet [22], VGG [23], GoogleNet [24], and ResNet [25] have been recognized as standards for object recognition tasks. The encoder part used for semantic segmentation is often built upon or

directly transferred from these architectures with variants in the decoder parts, such as SegNet [26], U-Net [27], and FCN [28]. Since most original architectures were proposed for sense parsing problems based on regular visible images, there is a lack of implementation in the delamination segmentation of the bridge deck based on thermal images. Thus, this study focuses on the fundamental build-up of the framework for field application instead of comparing the performance among different architectures.

2.3 Key components of DenseNet and DenseASPP

The architecture named DenseNet was introduced by Huang et al., (2017) and was selected as the encoder part of the proposed architecture. It was designed based on three key observations, which enhanced the performance of the model in the past: 1) connections from early layers to later layers; 2) concatenation of feature maps; 3) flow of information and gradients through the network. **Fig.2** illustrates an example of the dense block with four layers. It maximized the information flow by recursively connecting layers and concatenate feature maps. This was thought to be the key feature of the architecture [30]. Each layer in the dense block has a structure started at batch normalization (BN), rectified linear unit (ReLu), 1x1 convolution (Conv), and followed by BN-ReLu and 3x3 convolution [29]. After each layer, the channel-wise concatenation was conducted to fold feature maps from previous layers. After each dense block, a transition layer (T) was used to reduce the size of the feature map. We selected this architecture as the first part of the encoder.

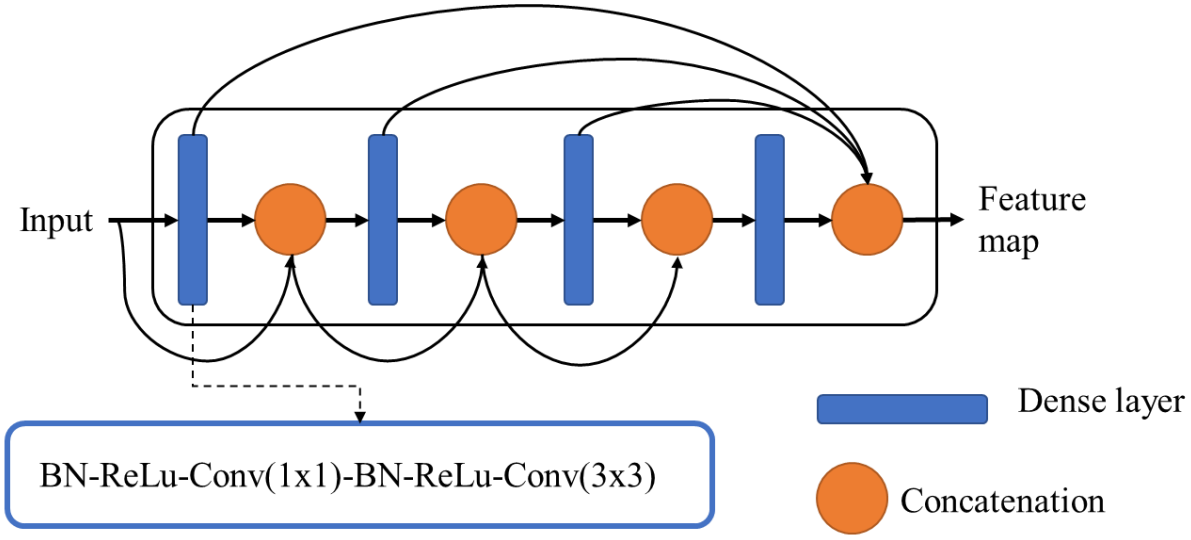

Figure 2. The diagram of Dense Block with k = 4 layers.

The densely connected atrous spatial pyramid pooling (DenseASPP) was proposed by Yang et al. (2018) to address the issue of multi-scale object encoding of image scene parsing for autonomous driving tasks. It consists of two key components: 1) dense connection inspired by DenseNet [29]; and 2) atrous spatial pyramid pooling [32,33]. **Fig.3** illustrates the structure of DenseASPP. The feature maps, often generated from the early layer, are encoded further by convolution with different dilation rates and then densely concatenated. The role of dilated convolution is to create different receptive fields, which is the key to capture multi-scale information of the object. We selected this module as the second part of the encoder due to its appropriateness for our case because delamination often occurs in different sizes and shapes.

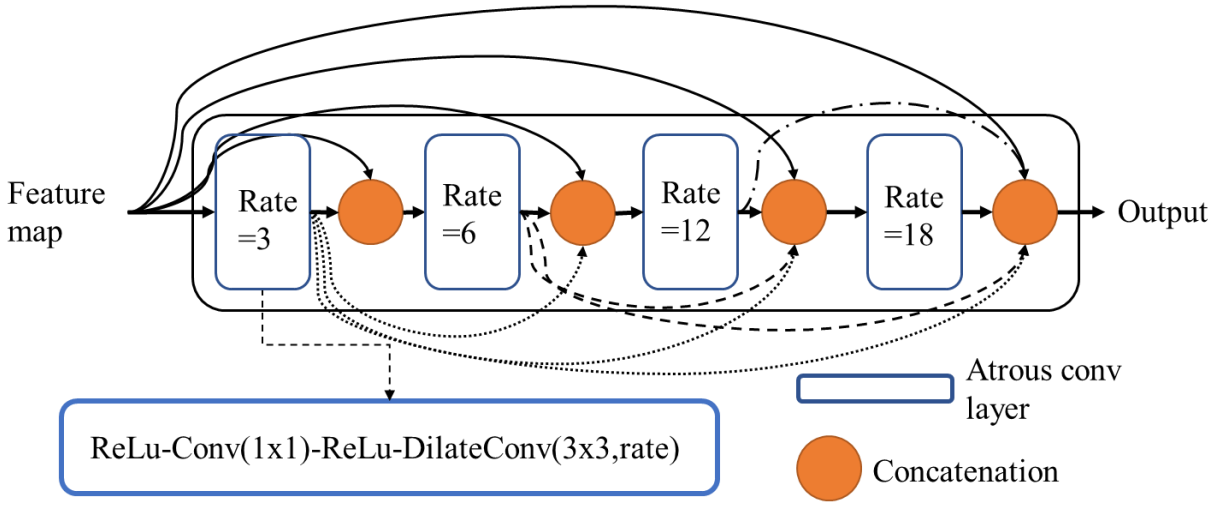

Figure 3. The diagram of DenseASPP with dilation rate of 3, 6, 12, and 18.

3 Methodology

3.1 Model Formation

The delamination profiling in a thermal image could be formatted in a general form of the segmentation problem. $T(i,j)$ is a thermal image and will be the input for the model (f_{model}). In each image $T(i,j)$, (*i,j*) refers to the temperature value at *ith* row and *j*th column. The expected output $P_c(i,j)$ is a probability map of the class *c* at location (*i,j*) of the image (**Eq. 2**). For our case, two classes are assigned to the model's prediction: delamination and non-delamination. Since the two classes have been assigned to the same location, the one-hot encoding (Dummy coding) is often used for multi-class representation [34]. Once the architecture of the model (f_{model}) has been built, a cost function (f_{cost} in **Eq. 3**) is used for model training under supervised learning scheme [21]. Since it follows the supervised learning framework, the labeled data Y_c is needed. The difference between the label Y_c and the prediction P_c is then measured by the loss function (f_{loss}). The training procedure is conducted by minimizing the cost function f_{cost} to fit the model given the input data.

$$f_{model}\big(T(i,j)\big) = P_c(i,j), \quad (2)$$

$$f_{cost} = f_{loss}(Y_c - P_c) \mid f_{model}, \quad (3)$$

$$minimize\ f_{cost}(\theta), \quad (4)$$

Where θ refers to the hyper-parameters such as learning rate, parameters for the loss function, etc.

3.2 Model Architecture

The architecture of the model f_{model} was shown in **Fig. 4,** which follows the encoder-decoder structure. The image as the input starts at the left side of the model, and the output is on the right side. The input image has a size of 256 by 256 pixel with 1 channel for temperature value. It passes through the first convolution layer (C1) consisting of a convolution filter with size 7 and 2 strides, followed by batch normalization, ReLu layer, and a max-pooling of 2 strides and results in 32 feature maps and a reduced dimension of 64x64 pixel size. Then the dense block (D1) with 4 layers and 16 filters in each layer was followed to enrich the feature maps up to 96 and a dimension reduction to 32 by the transition layer (T1), which includes a 1x1 convolution with 96 filters and an average pooling with 2 strides. The second dense block (D2) further expands the feature map to 256 with ten internal layers. A transition layer (T2) consisting of global batch normalization and dropout layer (DP) is followed to regularize the feature maps. Then the Dense ASPP layer remaps and expands the feature map into 256 with a dilation rate of 3, 6, 12, 18 for allocating multi-scale information. A compression layer (P) is used to reduce the number of feature maps from 256 to 128 by convolving with 3x3 filter. The decoder part consists of 3 up-sampling layers, which gradually increases the size of the feature

map and decreases the number of feature maps and finally retrieving the original image size with two channels. Each channel refers to the one-hot coding of the probability for each class P_c.

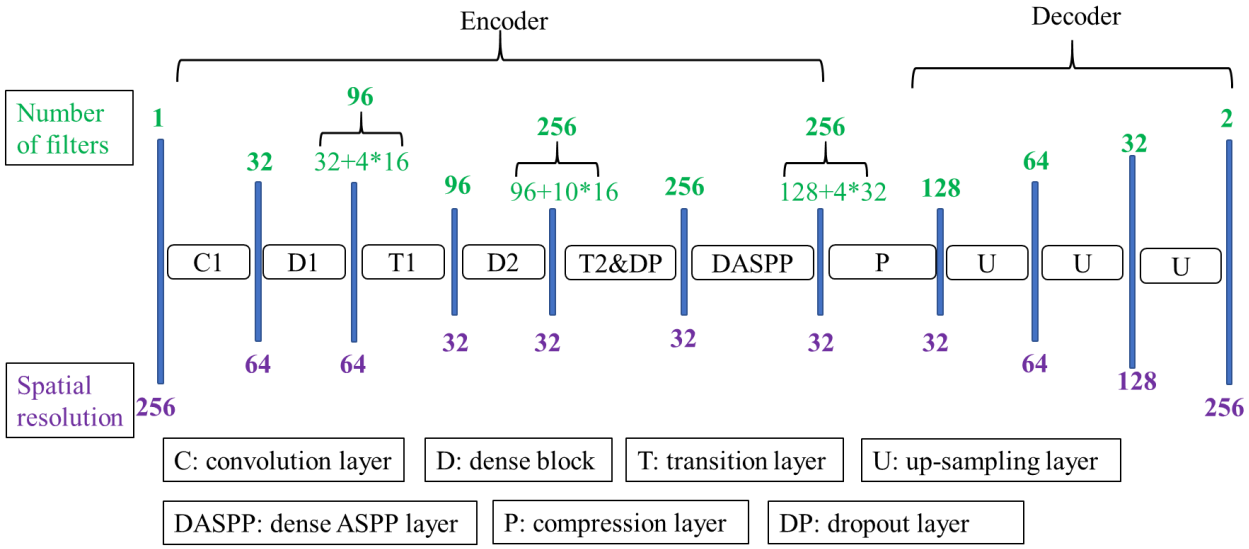

Figure 4. The diagram of the proposed architecture

3.3 Model Parameters and Training Settings

The loss function and the optimization algorithm are determined by the following a general selection from the literature [21,35,36]. Based on the histogram of training classes (**Fig.5c**), a large amount of imbalances in class distribution is observed where there were more numbers in the class of non-delamination ("0") than the ones in the class of delamination ("1") with the ratio around 8.6:1. With the imbalanced samples, the model would suffer low training accuracy [36]. Thus, the focal loss [36] was used to handle the imbalance with more focus on the class with fewer samples (defined in **Eq. 5**). There are two parameters in **Eq.5,** which $\alpha = 0.25$ and $\gamma = 2$ are used. The Adam optimization [35] was used to minimize the cost function with learning rate of 0.0005, and 0.9 and 0.999 for first and second momentum decay factors.

$$fl_{cost} = -\alpha(1-p_t)^{\gamma}\log(p_t), \text{ where } p_t = \begin{cases} p & if\ y = 1 \\ 1-p & otherwise \end{cases} \mid p \in P_c, y \in Y_c \quad (5)$$

Here, α refers to the balanced variant, and γ refers to the tunable focusing factor. Total iteration (epoch) of training was set to 30, and the mini-batch was deployed with the size of 24 [37].

3.4 Data Augmentation

The generalization ability is one of the key factors to determine the wellness of a model in terms of its robustness in the application. To have a good model, the dataset for training needs to be rich enough to cover the image variations in the population. For instance, ImageNet [38] has 1000 classes of labeled scenes with over 1 million images. The lack of a database of thermal images for delamination profiling in the current stage creates barriers for the transition of the state-of-art framework to the field implementation of NDT. To address the issue, the experimental study was conducted to collect the thermal images with three concrete slabs under the natural environment (See section 4.1 for detail). Under the experimental setup for the data collection, we identified that the collected data was still not representative of the practical condition in real-word scenarios which: 1) the size, shape, and location of mimicked delamination were fixed in the experiments but varied in real cases; 2) camera plan angle and zoom range always vary in real cases during data collection instead of the fixed height in the experiment. Thus, the data augmentation strategy, *random crop with automatic zoom* & *rotation*, was adopted to increase the sample variation to a more realistic situation (**Algorithm 1**):

Algorithm 1 Random crop with automatic zoom and rotation

For *i* in batch

$d_l = random(b_l, b_u)$	d_l: random length to be cropped
$d_d = random(0, 360)$	d_d: random degree between (0,360)
$x = random(0, w - d + 1)$	random start-point at x within image width w
$y = random(0, h - d + 1)$	random start-point at y within image height h
$I_{i\_crop} = I_i[\text{y: y + d, x: x + d}]$	cropped image I_{i_crop} from original image I_i
$I_{i\_resize} = interp(I_{i_{crop}}, [w, h])$	resize image to original size by interpolation
$M_r = rationMat(center, d_d)$	get rotation matrix at image center by degree of d_d
$I_{i\_rot} = affine(I_{i_{resize}}, M_r)$	apply rotation matrix to resized image I_{i_resize}

End for

The algorithm runs for the batch during the training following the uniform randomization and simultaneously processes the images and labels. The process of the random crop with automatic zoom aims to address the practical condition that the delamination is partially presented in the image with different scales. The random rotation fixes the issue for the situation that the camera or the surveyed bridge surface may be presented in any orientations. The only input for the algorithm is the lower bound b_l and upper bound b_u for the range of cropping in which $b_l = 90$ and $b_u = 255$ were used in the current study. **Fig.5a** illustrates a sample image collected from the experiment without augmentation. It shows the lowest variation of the sample in terms of fixed size and location under the experimental setup**. Fig.5b** and **c** show two samples after applying the augmentation in which the presences of delamination are varied by sizes and locations.

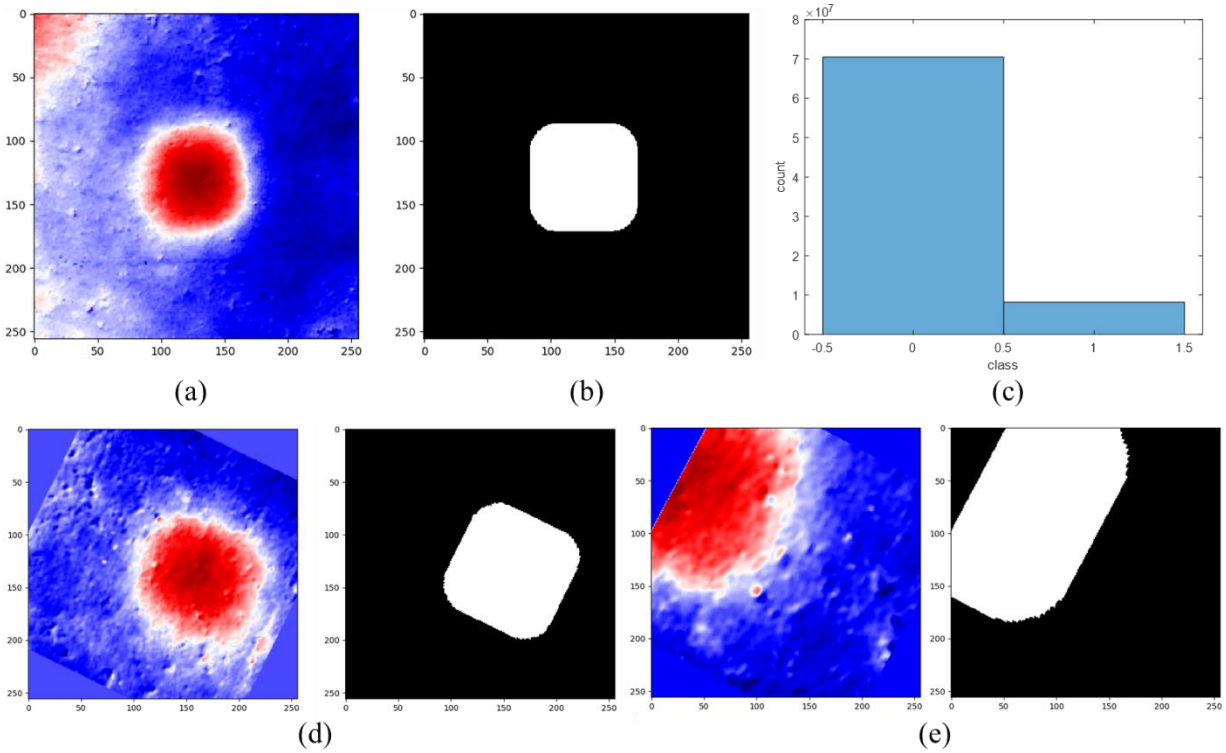

Figure 5. Sample data for raw and augmented images and labels: (a) sample image without augmentation; (b) corresponding label in (a); (c) class distribution for training ("0" for non-delamination and "1" for delamination); (d) sample 1 with augmentation and its label; (e) sample 2 with augmentation and its label.

3.5 Performance Evaluation Metrics

Since the outcome of the analysis is represented in the binary image, white regions in the image refer to the delamination (foreground) and the black regions to the non-delamination (background).

Several metrics can be used to evaluate the performance of the proposed architecture. Generally, it is defined as the comparison between the prediction by the algorithm and the ground-truths. Here, we adopt the intersection over union (*IOU*) in **Eq.6**, and area accuracy (**Eq.7**) to measure the accuracy of delamination segmentation. The *IOU* is a well-accepted metric measuring similarity between two sets and is often used for the performance evaluation in the image segmentation task [14]. *IOU* is also referred to as the Jaccard index, which quantifies the percent of the overlapped area between the prediction and the ground truth over the union of them [39]. This metric returns 0 if there is no overlapping and 1 if it is perfectly matched. It is also equivalent to true positive over the sum rates of true positive, false positive, and false negative, which accounts for both type I and type II errors. The area accuracy (R_{area}) defined in **Eq.7** is another indicator that calculates the rate of the detected area over the entire valid area. R_{area} is often used as a reference for maintenance decisions in practice, which is expected to be as close as possible to the ground truth. Precision and recall are often used metrics to give further understanding of prediction outcome. Precision, in our case, defines how many defected areas have been correctly detected given the ground truth (**Eq.8**). Recall measures the true positive among the positive detection (**Eq.9**).

$$IOU = \frac{|A_p \cap A_g|}{|A_p \cup A_g|} = \frac{TP}{TP+FP+FN} \quad (6)$$

$$R_{area} = \frac{A_p}{A_{total}} \quad (7)$$

$$Precision = \frac{TP}{TP+FP} \quad (8)$$

$$Recall = \frac{TP}{TP+FN} \quad (9)$$

Where in **Eq.6**, A_p is the predicted area by the algorithm, A_g is the ground truth area; $A_p \cap A_g$ calculates the intersection between prediction and ground truth, and $A_p \cup A_g$ returns the union area among prediction and ground truth. Also, TP refers to truth positive, FP refers to false positive, and FN refers to false negative. In **Eq.7**, A_{total} refers to the total surveyed area.

4 Experimental Study

4.1 Experimental Setup and Data Collection

Experimental studies were conducted by using the mimicked delamination in a reinforced concrete slab outdoors in sunny weather [16]. Three concrete slabs were cast. The layout of the slab is illustrated in **Fig. 6**. The Styrofoam, with a size of 25 cm by 25 cm, was buried at a depth of 4.5 cm, 7 cm, and 9.5 cm from the top surface of each slab (**Fig.6b**). The thickness of foam was measured about 0.4 cm with the thermal conductivity of 0.03 W/(m·K) in order to stimulate the delamination in the real bridge deck, which is identified as a thin layer of air with a thermal conductivity around 0.02 W/(m·K). The slabs were cured indoor over 28 days and then moved outside for the data collection. A thermal camera was used to collect the surface temperature data on multiple days in August and September 2018 (**Fig. 6c**).

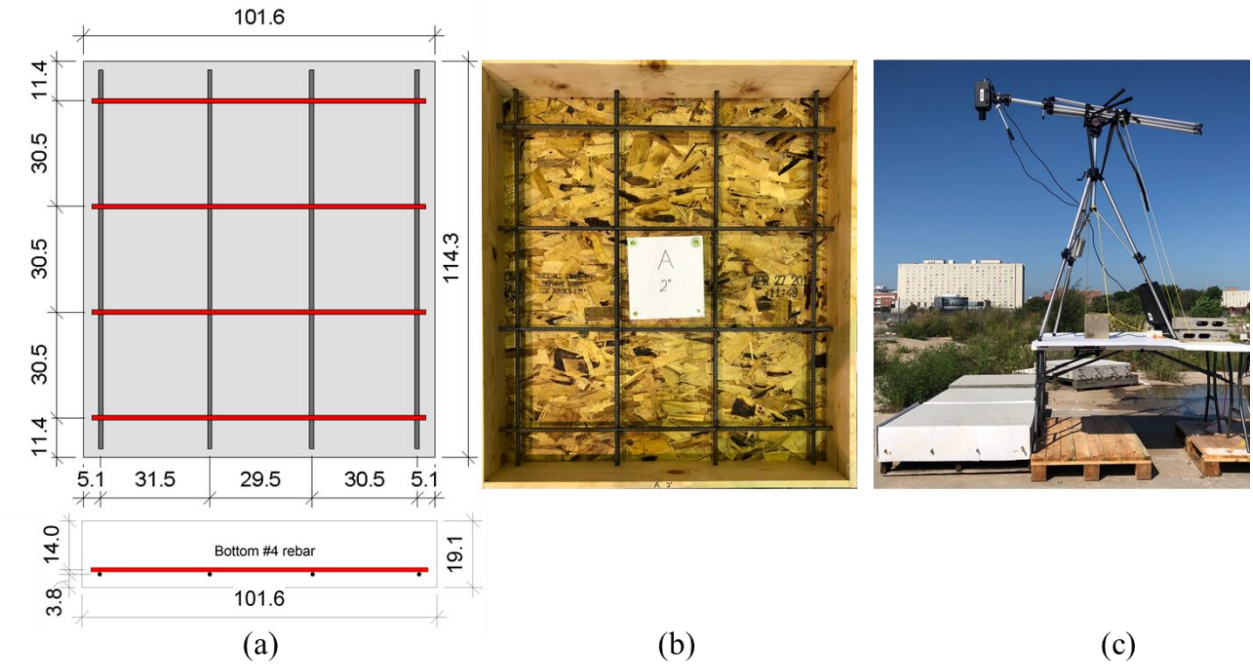

Figure 6. Experimental design, setup, and data collection

4.2 Datasets Partition and Comparison

After the data was collected, multiple datasets were designed to investigate the effects of the model performance and the sample varieties for model training. Each dataset has been divided into training and validation sets. Then, the model was trained and tested on different datasets. The training data consisted of three collections corresponding to the slabs with three depths in three days (**Table 1**: dataset 1, 2, and 3). Each dataset has 1200 thermal images recorded from 10:30 am to 5:30 pm of that day. Dataset 4 combined all cases of depths by down sampling each one by the rate of 3 and summing up to 1200 samples. Dataset 4, 5, and 6 were combined in pairs from dataset 1 to 3 in a similar way. Overall, 7 datasets had been generated for training to represent the conditions of different depths. The Testing data followed the same combination but was independently collected on different dates (**Table 1**). **Fig.7** shows that each dataset has been divided into a train-validate-test split. For each model, 80% of images are used for training and 20% for validation. To further test the model's generalization ability, the test data is from different independent experimentation. The similarity of the test dataset to the training dataset is assumed based on the similar weather condition. Based on the author's past research [40], similar weather conditions such as late-summer would provide the comparable thermographic response of the same defect. Although the testing dataset is comparable to the training set, variations due to the daily difference are expected. Testing the proposed model on handling these variations could furtherly evaluate the robustness of the framework.

Table 1. Dataset description and assignment

Dataset		**1**	**2**	**3**	**4**	**5**	**6**	**7**
Depth combination (cm)		4.5	7	9.5	4.5&7&9.5	4.5&7	4.5&9.5	7&9.5
Train	Temperature Range (°C)	29~46	25~38	26~37	25~46	25~46	26~46	25~38
	Dates	Day1	Day3	Day4	Day1,3,4	Day1,3	Day1,4	Day3,4
Validation	Temperature Range (°C)	31~46	32~45	27~46	27~46	31~46	27~46	27~46
	Dates	Day6	Day2	Day5	Day2,5,6	Day2,6	Day5,6	Day2,5

notes: Day 1(August 25), Day2 (August 27), Day3 (August 29), Day4 (September 10), Day5 (September 12), Day6 (September 17)

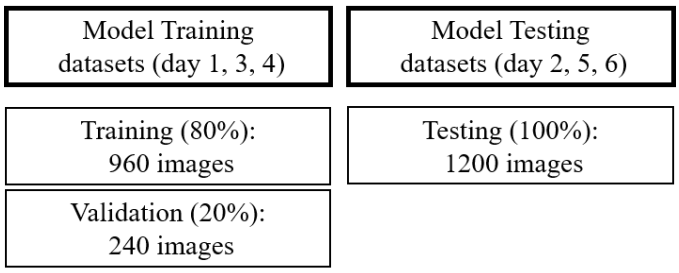

Figure 7. Data partition for training, validation, and testing

In addition to the dataset variation for depths and different time windows, the effect of augmentation was also compared (**Fig.8**). Thus, both training and testing datasets had two configurations: augmented and non-augmented. Cross evaluation was conducted where the models trained with/without augmentation were validated by the augmented and non-augmented validation and testing set. As a result, the effects of depths and augmentation could be fully evaluated. To focus on the evaluation of the effect of augmentation, the comparison is only made between 100% augmentation and 100% non-augmentation. The mixed training or testing for augmentation is not included in this study. In total, 14 models had been trained and evaluated based on the exact same architecture proposed in **section 3.1**.

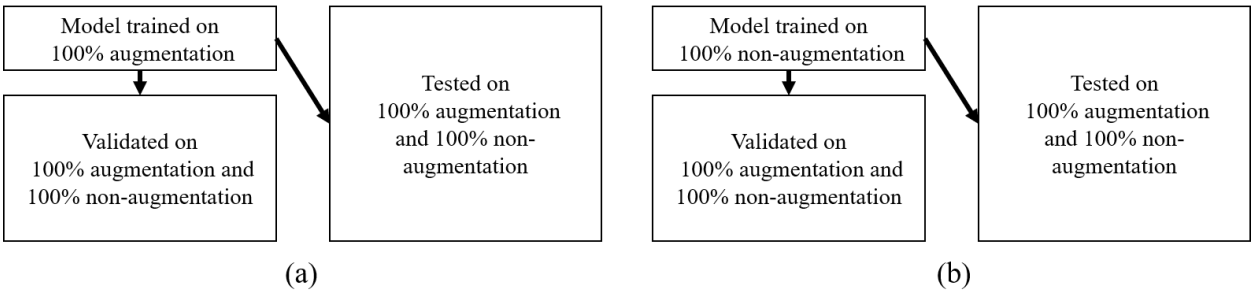

Figure 8. Evaluation of augmentation strategy: (a) model trained on an augmented dataset will be tested on both augmented and non-augmented validation and testing dataset; (b) model trained on a non-augmented dataset will be tested on both augmented and non-augmented validation and testing dataset;

4.3 Results Comparison by *IOU*

The training results, based on the *IOU* in **Table 2**, reveal that the proposed model is capable of learning and summarizing different datasets in terms of depth varieties and the simulated practical issues. Across the training sets for all datasets with and without augmentation, the models' prediction shows a consistent accuracy (99.77% for the non-augmented training and 88.36% for the augmented training). No significant difference was observed in training performance among datasets both in augmented and non-augmented training, which indicated the model was able to capture the thermal features of delamination at different depths. When comparing the training results averagely between the augmented and non-augmented training strategies, a decreased performance (~11%) was found in the augmented training. The training results also showed the models generally learned better in easy tasks (non-augmented training) than complicated tasks (augmented training).

Table 2. Training results comparison by *IOU* (%)

Dataset	1	2	3	4	5	6	7	Mean
Depth(cm)	4.5	7	9.5	4.5&7 &9.5	4.5&7	4.5& 9.5	7&9.5	
NON_AUG	99.86	99.87	99.88	99.58	99.70	99.73	99.75	**99.77**
AUG	84.85	91.64	90.57	85.98	88.65	88.45	85.14	**88.36**

notes: AUG: augmented training. NON_AUG: non-augmented training.

Table 3 showed the validation results for models using augmented and non-augmented training strategies across all datasets. For models trained on non-augmented and validated on non-

augmented strategies, all models return consistently high accuracy above 99%, which indicates models are well trained under the non-augmented setup. For models trained on augmented and validated on non-augmented strategies, the dataset 6 returned the highest accuracy (83.51%), and the dataset 1 returned the lowest accuracy (66.96%). For models trained on non-augmented and validated on augmented strategies, the dataset 1 returned the highest accuracy (53.22%), and the dataset 3 returned the lowest accuracy (15.43%). For models trained on augmented and validated by augmented strategies, dataset 2 returned the highest accuracy (91.86%), and the dataset 7 returned the lowest accuracy (84.88%). Based on the above comparison, the validation performance did not show a strong correlation to depth variations and not affected by the depth combination both in training and validation datasets, no matter if augmented or not. Instead, a significate performance degradation was observed when using a non-augmented trained model to predict the augmented validation dataset (from 88.93% to 33.19% in mean value). It is also observed that the model trained with augmentation performed a 10% difference in validation datasets between non-augmentation and augmentation (88.93% compared to 76.00% in mean value). It indicates that models with augmented training are more robust in spatial variations compared to the models with non-augmented training. When comparing the training and validation result, all models are well-trained corresponding to augmentation strategy (99.77% in **table 2** and 99.76% in **table 3**, 88.36% in **table 2** and 88.93% in **table 3**).

Table 3. Validation results comparison by *IOU* (%)

Dataset		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**Mean**
Depth(cm)		4.5	7	9.5	4.5&7 &9.5	4.5&7	4.5& 9.5	7&9.5	
Validation Non-AUG	NON_AUG	99.82	99.89	99.90	99.59	99.70	99.69	99.68	**99.76**
	AUG	66.96	83.45	73.42	76.84	71.81	83.51	74.54	**76.00**
Validation AUG	NON_AUG	53.22	25.91	15.43	30.79	40.07	33.70	25.29	**33.19**
	AUG	84.96	91.86	91.32	86.47	89.37	89.63	84.88	**88.93**

notes: AUG: augmented training. NON_AUG: non-augmented training.

Table 4 shows the results of models tested on dates other than training datasets. The results are used to further test the model's capability on handling variations in weather conditions due to the slight daily difference. The testing results agree to the validation results that no strong correlation exits in the model between depth variations. It also agrees that models perform more robust to spatial variations when trained with augmentation strategy. The major difference observed in the test dataset is that a general drop of *IOU,* ranging from 12% to 19%, is shown in the mean value of **table 4** from **table 3**. This indicates that the variations in daily difference could contribute to the degradation of performance for current models.

Table 4. Testing results comparison by *IOU* (%)

Dataset		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**Mean**
Depth(cm)		4.5	7	9.5	4.5&7 &9.5	4.5&7	4.5& 9.5	7&9.5	
Testing Non-AUG	NON_AUG	89.62	70.77	83.13	82.03	73.25	83.23	75.58	**80.34**
	AUG	68.96	55.30	66.68	68.69	62.84	63.48	69.48	**64.32**
Testing AUG	NON_AUG	57.67	29.00	17.61	39.24	46.15	40.76	31.96	**38.40**
	AUG	79.06	74.32	74.17	71.86	75.09	70.96	68.65	**74.24**

notes: AUG: augmented training. NON_AUG: non-augmented training.

Fig.9 shows sampled testing results from dataset 4 to reveal the visual investigation: 1) top left section shows the model's prediction on the augmented testing dataset with augmented training, where the most inaccuracy appears for the shape; 2) top right section shows the result on the non-augmented testing dataset with the augmented training, where the inaccuracy appears at the boundary; 3) bottom left section shows the results on the augmented testing dataset with the non-augmented training, where all predictions are smaller and tend to be located at the center of the image; 4) bottom right section shows the results on the non-augmented testing dataset with the non-augmented training,

which returns the highest accuracy. This means the model would profile the delamination in 82% accuracy to its real shape, given that the position of delamination is located in the middle of the image with different depths (training without augmentation). When the model was trained with augmentation, the model could profile the shape with 72% accuracy. It also found that the model could profile 39% accurate shape of the delamination given no constrain on size, location, and depth in the images (training without augmentation). This accuracy increased to 69% when the augmentation was introduced in training. In summary, when the model was trained by the data with high diversity (through augmentation), it would be capable of handling more situations than when it was trained by the data with low diversity.

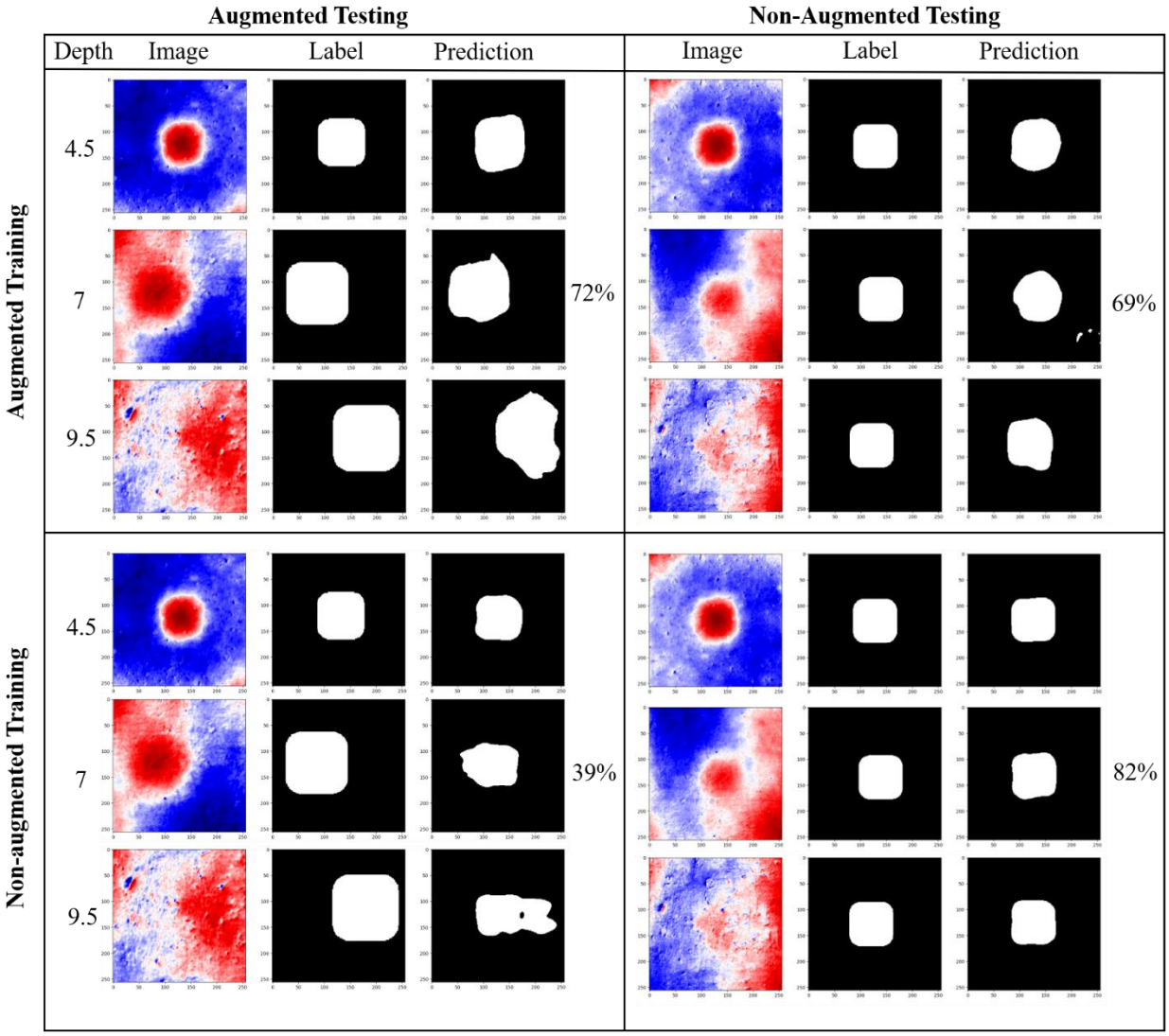

Figure 9. Testing results for dataset 4

4.4 Results Comparison by Precision and Recall

Besides the visual inspection of the prediction outcome, evaluating the precision and recall provides more insights on understanding the performance. Table 5 shows the mean precision and recall for training, validation, and testing results across all datasets. In the training process, both precision and recall return high value (larger than 90%) no matter if the augmentation is introduced. In the validation process, the precision decrease to 76.8% from 99.9% when the model validated on non-augmented dataset shifts training from non-augmentation to augmentation. During the validation with augmentation, the model drops in precision and recall significantly (65% and 37% compared to 92% and 95%) when evaluating the model with non-augmented training. In the testing process, the same trending is observed, but a drop in overall precision and recall are also presented. In summary, the model trained on the augmented dataset would decrease in precision when it is validated and tested on a non-augmented dataset. This means the false positive detection increased based on **Eq.8**. This is

also agreed with the visual hints in **Fig.9** top right section. When the model is trained on a non-augmented dataset and validated and tested on an augmented dataset, a significant decrease in precision and recall are observed. This means the model has high false positive and false negative detection.

Table 5. Mean precision and recall for training, validation, and testing results

	Training		Validation NON_AUG		Validation AUG		Testing NON_AUG		Testing AUG	
Model trained	NON_ AUG	AUG	NON_ AUG	AUG	NON_ AUG	AUG	NON_ AUG	AUG	NON_ AUG	AUG
Precision	99.88	91.98	99.88	76.80	65.23	92.27	87.53	67.78	68.14	84.30
Recall	99.89	95.29	99.87	98.35	37.47	95.48	89.20	93.41	42.52	85.02

notes: AUG: augmented training. NON_AUG: non-augmented training.

4.5 Results Comparison by Area Accuracy (R_{area})

The delaminated area percentage is another index that is often used as the reference for decision making to the appointment of the bridge maintenance. Thus, the accurate estimation of the delamination area percentage is desired. Depending on the selection of whether applying the augmentation, the area percentage of ground truth for augmented training and validation datasets was around 20.6%, and for non-augmented datasets was about 10.4%. **Table 6** shows the predicted area percentage differing from its corresponding ground-truth (label) for training datasets. Models from both training strategies (augmented and non-augmented) returned less than 1% difference in area percentage across all datasets. It indicated the prediction from models were not affected by the depth variations during the training.

Table 6. Area deviation from ground truth by training/validation results

Dataset	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**Mean**
Depth(cm)	4.5	7	9.5	4.5&7& 9.5	4.5&7	4.5& 9.5	7&9.5	
NON_AUG	0	0	0	0	0	0	0	0
AUG	+2.3	+0.2	+0.5	+0.3	+0.4	+0.7	+1.0	+0.8

*__notes__: 1) AUG: augmented training. NON_AUG: non-augmented training.

2) Area deviation $= R_{area}(\%) - groudtruth$

Table 7 shows the results of area accuracy predicted by validation datasets. For models trained on non-augmented and validated on non-augmented datasets, it returned 0% deviation for the area percentage. For models trained on augmented and validated on non-augmented datasets, dataset 2 returned the lowest deviation rate (1.3%), and dataset 1 returned the highest deviation rate of 5%. For models trained on non-augmented and validated on augmented datasets, dataset 2 returned the highest deviation (-10.9%), and the dataset 1 returned the lowest deviation (-7.2%). For models trained and validated on augmented datasets, dataset 2 returned the lowest deviations (-0.1%), and the dataset 1 returned the highest deviation (2.3%). There was no clear relationship found between the depth variation and its combination to the model's prediction in terms of area percentage. When comparing the average value of area percentage deviations across all datasets, the model with augmented training intended to predict more area by 3% on non-augmented validation datasets and return less than 1% increase of area on augmented validation datasets. The model with the non-augmented training returned about a 9% decrease in the area on the augmented validation dataset, which agreed with the low accuracy in terms of *IOU*. Interestingly, this amount of less prediction seems to close to the dataset whose area without augmentation and thus indicates that the model was "overfitted" to the training (low diversity) and performed poorly on the augmented validation (high diversity).

Table 7. Area deviation from ground truth by validation results

Dataset		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**Mean**
Depth(cm)		4.5	7	9.5	4.5&7 &9.5	4.5&7	4.5& 9.5	7&9.5	
Validation	NON_AUG	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Non-AUG	AUG	+5.0	+1.3	+3.1	+2.8	+3.7	+1.9	+3.3	+3.0
Validation	NON_AUG	-7.2	-10.9	-10.6	-9.4	-8.4	-9.2	-9.2	-9.3
AUG	AUG	+2.3	+0.1	+0.3	+0.4	+0.4	+0.7	+1.0	+0.8

*__notes__: 1) AUG: augmented training. NON_AUG: non-augmented training.

2) Area deviation $= R_{area}(\%) - groudtruth$

Table 8 shows the results of area accuracy predicted by testing datasets. For models trained on non-augmented and tested on non-augmented datasets, dataset 5 returned the highest deviation of 1.2% in area increase while dataset 1 returned 0.1% in area decrease. For models trained on augmented and tested on non-augmented datasets, dataset 3 returned the lowest deviation rate (2.3%), and dataset 6 returned the highest deviation rate of 5.6%. For models trained on non-augmented and validated on augmented datasets, the dataset 1 returned the lowest deviation (-7.6%), and the dataset 4 returned the highest deviation (-10.5%). For models trained and validated on augmented datasets, the dataset 5 returned lowest deviations (-0.2%), and the dataset 2 returned the highest deviation (-2.4%). There was no clear relationship found between the depth variation and its combination to the model's prediction in terms of area percentage. The testing results agreed to the validation results by the mean values, which supported the usefulness of augmentation strategy in the outcome of area accuracy.

Table 8. Area deviation from ground truth by testing results

Dataset		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**Mean**
Depth(cm)		4.5	7	9.5	4.5&7 &9.5	4.5&7	4.5& 9.5	7&9.5	
Testing Non-AUG	NON_AUG	-0.1	-0.2	+0.9	-0.3	+1.2	+0.3	+0.2	+0.3
	AUG	+4.4	+5.2	+2.3	+3.6	+5.4	+5.6	+4.1	+4.4
Testing AUG	NON_AUG	-7.6	-8.8	-9.5	-10.5	-8.1	-8.8	-9.9	-9.0
	AUG	+0.7	-2.4	-1.6	-1.0	-0.2	+0.9	+0.5	-0.5

*__notes__: 1) AUG: augmented training. NON_AUG: non-augmented training.

2) Area deviation $= R_{area}(\%) - groudtruth$

5 Field Implementation

5.1 Data Collection and Image processing

An in-service concrete bridge with the deck overlay was surveyed in October 2017 at US 77 close to Lincoln, Nebraska. **Fig.10** shows the field-setup for data collection. Two UAV configurations were employed to collect the visible and thermal images of the deck surface. The thermal camera FLIR A8300 was mounted on the DJI Matrice 600 with customized design to ensure the orthogonal field of view. The camera was controlled by an on-board computer for data storage. The UAV was controlled by a licensed remote pilot manually and followed the center of the bridge at a constant speed. The raw images were then processed through the customized MATLAB algorithms for correction and stitching. Then the quality of the stitched thermal image of the bridge was manually checked with the visible image in this case.

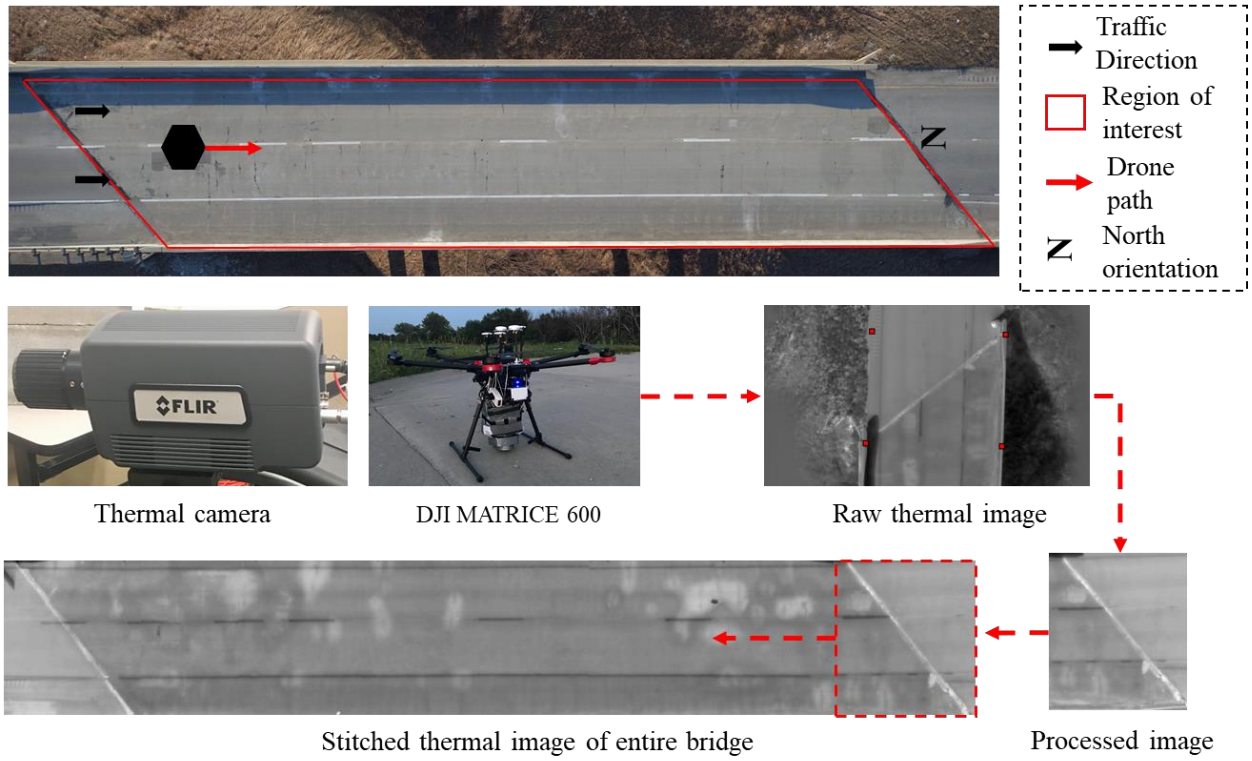

Figure 10. Field step, data collection, and processing

5.2 Sliding Window Detector for Bridge Implementation

After the model was trained, and the thermal images of the bridge were processed using **Eq. 1**, a sliding window detector was proposed to apply the model prediction for the bridge implementation (**Fig.11**). First, a hammer sounding result provided by the Nebraska Department of Transportation (NDOT) was used as the reference to help compare the outcome predicted by the model. It is a conventional method for shallow delamination profiling on-site (ASTM 4850M-12), and the boundaries of delamination were directly marked with chalk lines while hammering. In addition to the hammer sounding, 12 coring samples were taken on the bridge deck, which furtherly validated the hammer sounding result [16]. Here, we overlapped the hammer sounding result on the raw image shown in **Fig.11a** (blue lines). Then, a *sliding window detector* was designed to pass through each lane of the bridge from one side to another with a fixed interval (red box in **Fig.11a**). The size of the interval defines how much overlapping between two successive windows. Based on the author's experimentation, it was found that in the model's prediction, the middle strip in the sliding window often returns the most stable outcome in terms of shape (**Fig.11b**). Thus, the only prediction from the middle strip of each sliding window was used, and the final output was stitched sequentially from each strip. The procedure is shown in the following **algorithm2**:

Algorithm 2 *Sliding window detector*

Define overlapping ratio r, desired image size (l)
Rescale $I(r,c) \rightarrow I'(r',c')$, rescale the window size to the desired image size (l)
$d \leftarrow (1-r) * l$, calculate the interval d
For i in range $(1, \frac{r'}{d})$: in the range of total number of sliding windows
    $s_i = I'(:, i: i+l)$, extract each window from rescaled raw thermal image I'
    $p_i = model(s_i)$ get prediction from trained model given each s_i
    If $i * l > l/2$:
        $ouput(i) = p_i(:, \frac{l}{2} : \frac{l}{2} + d)$ extract middle strip for the final output
    End if
    $final = append[final, output(i)]$ sequentially stitch for final output
End for

Based on this procedure, a small interval of moving was required and resulted in a high overlapped area (90% in this paper). The final output is shown in **Fig.11c**, which is a delamination probability pulled out from the softmax layer of the model at each location across the bridge deck (1 indicates delamination and 0 indicates non-delamination). Here, we used the model trained on dataset 4 with augmentation for illustration. **Fig.11d** shows the result comparison between hammer sounding and our approach. It is found that our approach (area ratio of 20.2%) predicted more regions than the hammer soundings (area ratio of 12.6%) as potential delamination. However, it is also found that the results indicated by hammer sounding were majorly included by our approach.

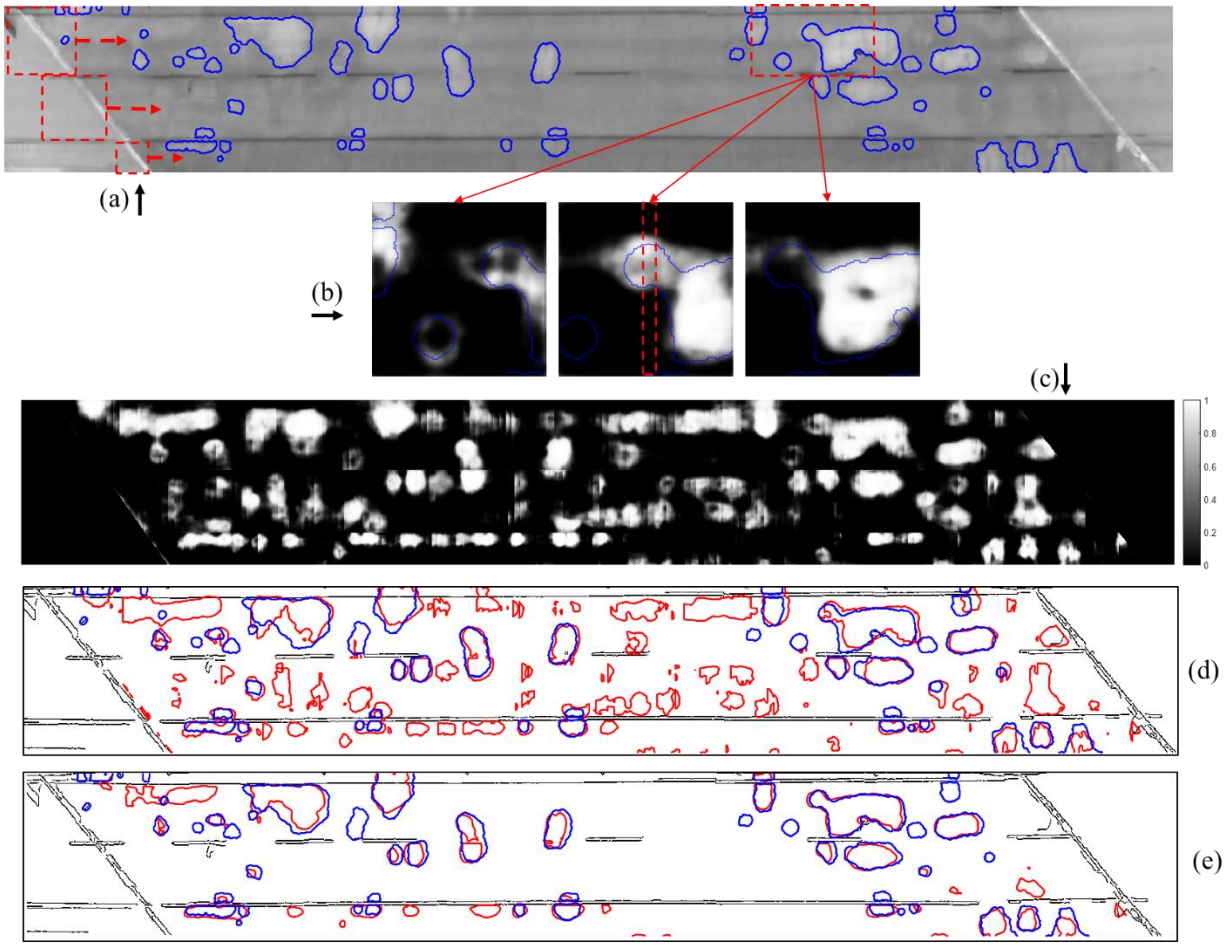

Figure 11. Field implementation: (a) procedure of sliding window detector (red box) on the bridge with hammer sounding result overlapped (blue color); (b) intermediate result of the detector showing that the middle strip returned most stable prediction; (c) delamination probability prediction generated from model's softmax layer; (d) final result by model prediction (red) and compared to hammer sounding (blue); (e) refinement of (c) by CRF and compared to hammer sounding outcomes

5.3 Refinement by Conditional Random Field (CRF)

An often-used post-processing technique in deep learning to refine the model prediction was to infer the posterior distribution in a conditional random field [41]. CRF refines the distribution given predictions from the model and raw image features. In our case, we followed the approach proposed by Krähenbühl and Koltun (2011), which was based on the Gaussian kernel for edge potential approximation. **Fig.11e** shows the refined outcome from **Fig.11d** and the raw thermal image in **Fig.10**. CRF smoothed and refined the boundaries from the trained model's direct predictions and removed small regions. As a result, CRF returned less area percentage deviation (-2.2%) from hammer sounding compared to the model's direct prediction (+6.4%). It also improved the segmentation accuracy by 15% compared to the model's direct prediction. Compared to the outcome predicted by the model that was trained on same dataset but without augmentation, the performance was significantly degraded in both visual clues and quantitative measures. **Fig.12a** shows the probability result of the model prediction exhibiting a clear clue of overfitted model, where three horizontal white bands cross

the whole image. This behavior occurred at models for all datasets (except dataset 1) and thus returned poor performance in terms of area percentage and accuracy (**Fig.12b**). Also, post-processing with CRF did not improve the overfitted model (**Fig.12c**).

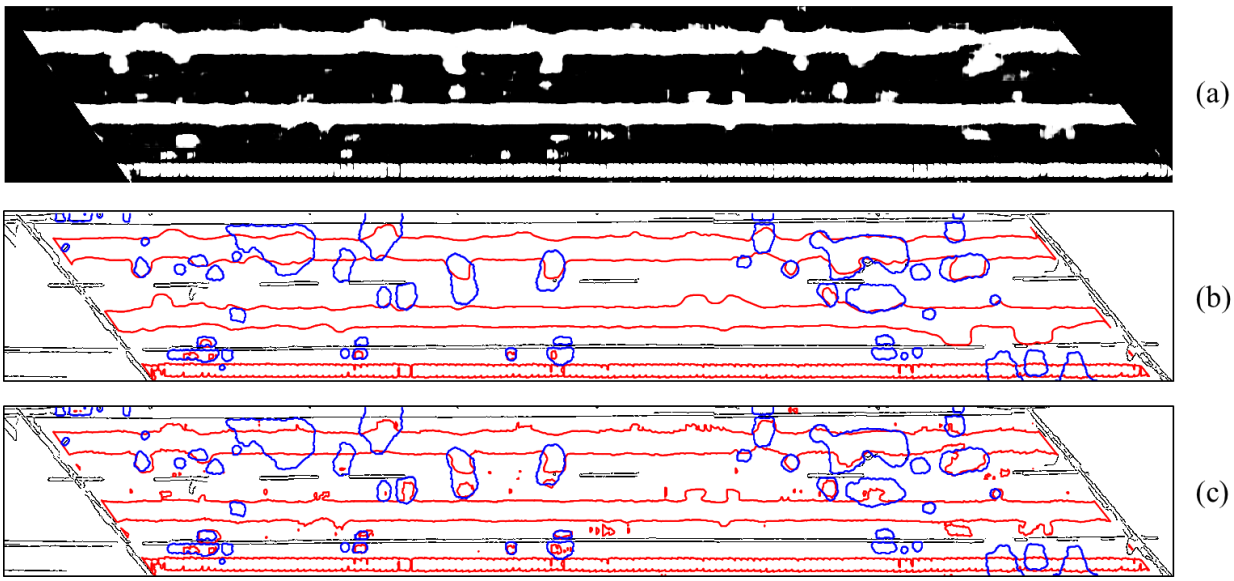

Figure 12. Model prediction based on non-augmented training for dataset 4: (a) probability display of model's prediction; (b) result comparison between model's direction prediction (red) and hammer sound (blue); (c) result comparison between model's prediction after CRF (red) and hammer sounding (blue)

5.4 Discussion of Different Models

Across models trained from different datasets, the augmentation had a significant effect on the model's generalization ability for the implementation in the real-world scenario. **Table 9** shows the *IOU* accuracies, area deviations, precision, and recall of models with and without CRF post-processing. Here, we used the hammer sounding result as the reference. Within the results of direct prediction of models, accuracy was improved, and deviation of area percentage was reduced when comparing the augmented training to non-augmented training (around 10% average reduction in area percentage deviation and 20% improvement for accuracy). Within the results post-processed by CRF, the accuracy was increased by 30%, and the deviation of area percentage was reduced by 10% when comparing the augmented training to non-augmented training. Among the models trained with augmentation, the CRF improved the accuracy of segmentation by 14% and reduced the deviation of area percentage by 6%. In terms of precision and recall, augmentation increased both values, which indicates a general improvement. CRF improved precision significantly in the augmented model, which supported the removal of false-positive detection. CRF did not improve the recall, which means the model missed some areas. Among models with different datasets in depths, there was no significant difference in performance when combining multiple depths sample than the dataset of single depth for training.

It is noticeable that the current overall model accuracy was lower in the bridge implementation than the ones in the experimental studies. The best result is 51% *IOU* by the model trained with augmentation after CRF. Generally, the *IOU* is larger than 50% can be treated as "good" in objective detection tasks [42]. Here we think this threshold is appropriate given that the absolute ground truth rather than hammer sound is hard to have. With reviewing the precision and recall of the model's prediction on the bridge in **Table 9,** we found that our model can correctly detect the defected area among the ground truth (69.61% in recall before CRF). On the other hand, the model might tend to miss classify the sound area as the defected area (72.41% in precision after CRT). In response to the miss of detection, we found the thermal signal may be disappeared if there the water invaded the delaminated layer. As a result, the hot pattern is no longer feasible for a model to detect. In terms of false detection, we found that some foreign dark color spots (caused by sticky asphalt or other external dark materials) may cause false-positive results. In general, the developed method has limitations in handling special environmental noises (such as uneven dark colors, artificial heat/cold sources, and

shadows etc.), which can fool the developed algorithm and cause inaccurate predictions. However, for the concrete bridge decks under normal operations and maintenance, our method proved to be effective and efficient.

Table 9. *IOU* (%), area accuracy (%), precision (%) and recall (%) for bridge implementation

Model Trained		NON-AUG	AUG
Direct Prediction	*IOU*	15.8	**36.9**
	Area Deviation	17.6	**7.4**
	Precision	21.09	**44.01**
	Recall	42.14	**69.61**
Prediction with CRF	*IOU*	18.5	**51.0**
	Area Deviation	11.9	**1.8**
	Precision	32.62	**72.41**
	Recall	37.51	**63.54**

*__notes__: 1) AUG: augmented training. NON_AUG: non-augmented training.

2) Area deviation $= R_{area}(\%) - groudtruth$

Instead of limitations of the proposed model, using the hammer sounding result as "ground truth" may not be optimal. Although hammer sounding is a trusted method for shallow delamination detection for the bridge deck, the condition of the bridge deck is more complicated; often, multiple types of defects existed mutually. Besides, some minor shifts occurred when aligning the hammer sounding result to the thermographic result. Thus, a practical evaluation of the bridge deck often involved multiple measurements based on different NDT methods [5]. Currently, the inference of thermography for bridge deck evaluation is still an active topic in the field of NDT, which requires more attention in the community. On the other side, limited training samples with low diversity and under-development of the technology implementation are other factors constrained the current state of the art.

6 Conclusions

This paper presented a framework to utilize supervised deep learning for automating delamination profiling using thermography for real-world applications based on limited experimental image data. The results of the experimental study and the practical implementation of this framework demonstrated the capability of this proposed approach, which provided a way to push the envelope of infrared based NDT in concrete delamination segmentation. The paper addressed two challenges in the current state-of-the-practice: the training factors affecting model performance and the implementation of the bridge case for automatic processing. The experimental study reveals that data augmentation strategies such as random crop, zoom, and rotation played significant effects on the model's generalization ability. It was found that introducing the augmentation would decrease by around 10% training accuracy but significantly increase the accuracy by over 35% in validation and testing datasets when mimicking the random condition in the validation dataset. The model tends to be overfitted when no augmentation is introduced in the training data. This observation was later supported by the real-world implementation on an in-service concrete bridge. Under further evaluation by precision and recall, it reveals that the model intended to falsely predict defected areas when it was trained on augmented data and validated and tested on non-augmented data. On the other hand, the model suffered both low precision and recall when it was trained on non-augmented data and tested on augmented data. This supported the importance of the diversity of training data and, thus, the usefulness of the augmentation strategy.

The fully automatic processing of the thermal image of the real bridge was achieved through the proposed algorithm once the model was trained by experimental data. The result comparison of different training datasets suggests that proper image augmentation strategies are necessary for improving the accuracy and robustness of the model. The post-processing method, conditional random field, could be used to further improve the performance. The overall performance of the model for field implementation can be improved by including more real-world data in the training

process in the future. Given the fact that most of today's delamination data processing is largely handled by manual methods, being able to automate such a task means significant cost-reduction potential in bridge maintenance without the constraints of traffic closure, which are typically needed in traditional deck inspections. It also means great quality-enhancement potential by conducting more frequent aerial inspections to acquire more updated bridge condition data.

Several limitations also need to be addressed in future works. The current study focused on the applicability of the deep learning model in the encoder-decoder architecture. Other architectures existed for visual optical segmentation need further evaluations in terms of performance improvement. Also, the study aimed to testify the effectiveness of the augmentation strategy, the effectiveness of mixed training strategy (including both augmented and non-augmented datasets) requires further evaluation.

ACKNOWLEDGMENT

The authors would like to thank the Nebraska Department of Transportation for their efforts in facilitating the data collection and sharing their non-destructive evaluation results.

7 References

[1] Lin W, Yoda T. Bridge engineering: Classifications, design loading, and analysis methods. 2017.

[2] Gucunski N, Imani A, Romero F, Nazarian S, Yuan D, Wiggenhauser H, et al. Nondestructive Testing to Identify Concrete Bridge Deck Deterioration. 2012. https://doi.org/10.17226/22771.

[3] Abu Dabous S, Yaghi S, Alkass S, Moselhi O. Concrete bridge deck condition assessment using IR Thermography and Ground Penetrating Radar technologies. Autom Constr 2017. https://doi.org/10.1016/j.autcon.2017.04.006.

[4] Kee SH, Oh T, Popovics JS, Arndt RW, Zhu J. Nondestructive bridge deck testing with air-coupled impact-echo and infrared thermography. J Bridg Eng 2012. https://doi.org/10.1061/(ASCE)BE.1943-5592.0000350.

[5] Lin S, Meng D, Choi H, Shams S, Azari H. Laboratory assessment of nine methods for nondestructive evaluation of concrete bridge decks with overlays. Constr Build Mater 2018. https://doi.org/10.1016/j.conbuildmat.2018.08.127.

[6] Maierhofer C, Arndt R, Röllig M, Helmerich R, Walther A, Hillemeier B, et al. Quantification of Voids and Delaminations in Real Concrete and Masonry Structures with Active Thermography: Case Studies, 2006. https://doi.org/10.21611/qirt.2006.049.

[7] Washer G, Fenwick R, Nelson S, Rumbayan R. Guidelines for thermographic inspection of concrete bridge components in shaded conditions. Transp Res Rec 2013. https://doi.org/10.3141/2360-02.

[8] Oh T, Kee SH, Arndt RW, Popovics JS, Zhu J. Comparison of NDT methods for assessment of a concrete bridge deck. J Eng Mech 2013. https://doi.org/10.1061/(ASCE)EM.1943-7889.0000441.

[9] Omar T, Nehdi ML. Clustering-Based Threshold Model for Condition Assessment of Concrete Bridge Decks Using Infrared Thermography, 2018. https://doi.org/10.1007/978-3-319-61914-9_19.

[10] Abdel-Qader I, Yohali S, Abudayyeh O, Yehia S. Segmentation of thermal images for non-destructive evaluation of bridge decks. NDT E Int 2008. https://doi.org/10.1016/j.ndteint.2007.12.003.

[11] Ellenberg A, Kontsos A, Moon F, Bartoli I. Bridge deck delamination identification from unmanned aerial vehicle infrared imagery. Autom Constr 2016. https://doi.org/10.1016/j.autcon.2016.08.024.

[12] Zhu H, Meng F, Cai J, Lu S. Beyond pixels: A comprehensive survey from bottom-up to semantic image segmentation and cosegmentation. J Vis Commun Image Represent 2016. https://doi.org/10.1016/j.jvcir.2015.10.012.

[13] Garcia-Garcia A, Orts-Escolano S, Oprea S, Villena-Martinez V, Martinez-Gonzalez P, Garcia-Rodriguez J. A survey on deep learning techniques for image and video semantic segmentation. Appl Soft Comput J 2018. https://doi.org/10.1016/j.asoc.2018.05.018.

[14] Yu H, Yang Z, Tan L, Wang Y, Sun W, Sun M, et al. Methods and datasets on semantic segmentation: A review. Neurocomputing 2018. https://doi.org/10.1016/j.neucom.2018.03.037.

[15] Cheng C, Na R, Shen Z. Thermographic Laplacian-pyramid filtering to enhance delamination detection in concrete structure. Infrared Phys Technol 2019.

https://doi.org/10.1016/j.infrared.2018.12.039.

[16] Cheng C, Shang Z, Shen Z. Bridge deck delamination segmentation based on aerial thermography through regularized grayscale morphological reconstruction and gradient statistics. Infrared Phys Technol 2019. https://doi.org/10.1016/j.infrared.2019.03.018.

[17] Omar T, Nehdi ML, Zayed T. Infrared thermography model for automated detection of delamination in RC bridge decks. Constr Build Mater 2018. https://doi.org/10.1016/j.conbuildmat.2018.02.126.

[18] Cheng C, Shen Z. Detecting Concrete Abnormality Using Time-series Thermal Imaging and Supervised Learning. ArXiv Prepr ArXiv180405406 2018.

[19] Büyüköztürk O. Imaging of concrete structures. NDT E Int 1998. https://doi.org/10.1016/S0963-8695(98)00012-7.

[20] LeCun YA, Bottou L, Orr GB, Müller K-R. Efficient BackProp BT - Neural Networks: Tricks of the Trade. Neural Networks: Tricks of the Trade, 2012.

[21] Lecun Y, Bengio Y, Hinton G. Deep learning. Nature 2015. https://doi.org/10.1038/nature14539.

[22] Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Adv. Neural Inf. Process. Syst., 2012.

[23] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. ArXiv Prepr ArXiv14091556 2014.

[24] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, et al. Going deeper with convolutions. Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., 2015. https://doi.org/10.1109/CVPR.2015.7298594.

[25] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., 2016. https://doi.org/10.1109/CVPR.2016.90.

[26] Badrinarayanan V, Kendall A, Cipolla R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. IEEE Trans Pattern Anal Mach Intell 2017. https://doi.org/10.1109/TPAMI.2016.2644615.

[27] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), 2015. https://doi.org/10.1007/978-3-319-24574-4_28.

[28] Long J, Shelhamer E, Darrell T. Fully Convolutional Networks for Semantic Segmentation. IEEE Trans Image Process 2014. https://doi.org/10.1109/IJCNN.2017.7966367.

[29] Huang G, Liu Z, Van Der Maaten L, Weinberger KQ. Densely connected convolutional networks. Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017, 2017. https://doi.org/10.1109/CVPR.2017.243.

[30] Jegou S, Drozdzal M, Vazquez D, Romero A, Bengio Y. The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work., 2017. https://doi.org/10.1109/CVPRW.2017.156.

[31] Yang M, Yu K, Zhang C, Li Z, Yang K. DenseASPP for Semantic Segmentation in Street Scenes. Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., 2018. https://doi.org/10.1109/CVPR.2018.00388.

[32] Chen LC, Papandreou G, Kokkinos I, Murphy K, Yuille AL. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. IEEE Trans Pattern Anal Mach Intell 2018. https://doi.org/10.1109/TPAMI.2017.2699184.

[33] Chen L-C, Zhu Y, Papandreou G, Schroff F, Adam H. Rethinking Atrous Convolution for Semantic Image Segmentation Liang-Chieh. ArXivOrg 2018. https://doi.org/10.1159/000018039.

[34] Garavaglia S, Sharma A, Hill M. a Smart Guide To Dummy Variables : Four Applications and a Macro. Entropy 1998.

[35] Kingma DP, Ba JL. Adam: A method for stochastic gradient descent. ICLR Int Conf Learn Represent 2015.

[36] Lin TY, Goyal P, Girshick R, He K, Dollar P. Focal Loss for Dense Object Detection. Proc. IEEE Int. Conf. Comput. Vis., 2017. https://doi.org/10.1109/ICCV.2017.324.

[37] Li M, Zhang T, Chen Y, Smola AJ. Efficient mini-batch training for stochastic optimization. Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., 2014. https://doi.org/10.1145/2623330.2623612.

[38] Jia Deng, Wei Dong, Socher R, Li-Jia Li, Kai Li, Li Fei-Fei. ImageNet: A large-scale hierarchical image database, 2009. https://doi.org/10.1109/cvprw.2009.5206848.

[39] Wang S. New benchmark for image segmentation evaluation. J Electron Imaging 2007.

https://doi.org/10.1117/1.2762250.

[40] Cheng C, Shen Z. The application of gray-scale level-set method in segmentation of concrete deck delamination using infrared images. Constr Build Mater 2020;240:117974.

[41] Krähenbühl P, Koltun V. Efficient inference in fully connected crfs with Gaussian edge potentials. Adv. Neural Inf. Process. Syst. 24 25th Annu. Conf. Neural Inf. Process. Syst. 2011, NIPS 2011, 2011.

8 Appendix

Table I. Precision and recall for training datasets

Dataset (training)		1	2	3	4	5	6	7	Mean
Depth(cm)		4.5	7	9.5	4.5&7&9.5	4.5&7	4.5&9.5	7&9.5	
Training NON_AUG	precision	99.93	99.93	99.94	99.79	99.84	99.86	99.88	99.88
	recall	99.93	99.94	99.94	99.80	99.85	99.87	99.88	99.89
Training AUG	precision	87.28	95.17	94.07	91.94	93.05	92.31	90.01	91.98
	recall	96.94	96.14	96.09	93.13	95.05	95.55	94.10	95.29

Table II. Precision and recall for validation datasets

Dataset (validation)			1	2	3	4	5	6	7	Mean
Depth(cm)			4.5	7	9.5	4.5&7&9.5	4.5&7	4.5&9.5	7&9.5	
Validation Non-AUG	NON_AUG	precision	99.91	99.93	99.96	99.87	99.90	99.85	99.77	99.88
		recall	99.91	99.95	99.94	99.72	99.80	99.85	99.91	99.87
	AUG	precision	67.15	85.68	75.12	77.74	72.75	84.05	75.11	76.80
		recall	99.58	96.95	97.01	98.52	98.12	99.24	99.04	98.35
Validation AUG	NON_AUG	precision	86.66	57.40	38.19	70.30	75.87	70.41	57.80	65.23
		recall	58.07	32.13	20.58	35.41	45.95	39.18	31.00	37.47
	AUG	precision	87.91	96.20	95.03	90.71	93.24	93.72	89.08	92.27
		recall	96.31	95.34	95.92	94.96	95.62	95.41	94.83	95.48

Table III. Precision and recall for testing datasets

Dataset (testing)			1	2	3	4	5	6	7	Mean
Depth(cm)			4.5	7	9.5	4.5&7&9.5	4.5&7	4.5&9.5	7&9.5	
Testing Non-AUG	NON_AUG	precision	95.24	83.46	87.41	91.23	80.40	89.91	85.02	87.53
		recall	93.77	82.24	94.38	88.53	87.06	91.59	86.83	89.20
	AUG	precision	69.52	59.76	72.73	71.69	64.95	64.91	70.94	67.78
		recall	98.50	87.97	86.85	94.55	91.87	96.80	97.30	93.41
Validation AUG	NON_AUG	precision	90.68	57.06	37.20	75.63	79.41	72.95	64.03	68.14
		recall	61.86	35.61	23.67	42.19	51.60	46.13	36.58	42.52
	AUG	precision	85.61	89.57	86.60	84.00	84.93	80.15	79.27	84.30
		recall	92.37	81.05	82.46	83.33	86.70	86.03	83.22	85.02

Table IV. *IOU* (%) and area accuracy (%) for bridge implementation by different models

Model Trained by Dataset			**1**	**2**	**3**	**4**	**5**	**6**	**7**	**Abs Mean**
Direct Prediction	IOU	NON_AUG	34.4	10.5	6.6	15.1	15.8	14.1	14.1	15.8
		AUG	38.1	37.2	32.7	39.8	40.2	36.3	34.1	**36.9**
	Area Deviation	NON_AUG	-1.4	+7.9	+18.4	+22.1	+24.7	+24.2	+24.2	17.6
		AUG	+7.1	+8.9	+6.8	+6.4	+5.1	+9.0	+8.4	**7.4**
Prediction with CRF	IOU	NON_AUG	34.8	9.1	5.3	15.1	16.5	33.4	15.3	18.5
		AUG	49.7	48.0	49.0	54.6	52.2	55.0	48.5	**51.0**
	Area Deviation	NON_AUG	-7.7	-7.2	+8.0	+17.5	+20.8	+5.5	+16.7	11.9
		AUG	-1.5	+1.2	-2.2	-2.2	-2.8	-1.6	-1.1	**1.8**

Table V. *Precision* (%) and recall (%) for bridge implementation by different models

Model Trained by Dataset			1	2	3	4	5	6	7	Abs Mean
Direct Prediction	NON_AUG	precision	54.22	15.41	8.77	17.85	18.21	16.57	16.57	21.09
		recall	48.27	25.03	21.60	49.18	53.96	48.46	48.46	42.14
	AUG	precision	45.20	42.97	40.66	47.30	49.10	42.18	40.67	**44.01**
		recall	70.77	73.43	62.59	71.50	68.88	72.43	67.69	**69.61**
Prediction with CRF	NON_AUG	precision	92.68	27.90	8.16	18.65	19.47	42.50	18.96	32.62
		recall	35.77	11.91	13.40	44.58	51.70	61.06	44.18	37.51
	AUG	precision	70.80	62.04	72.82	77.93	78.51	76.11	68.64	**72.41**
		recall	62.56	67.92	59.94	64.53	60.94	66.53	62.37	**63.54**