NB-CNN: Deep Learning-Based Crack Detection Using Convolutional Neural Network and Naïve Bayes Data Fusion

Fu-Chen Chen and Mohammad R. Jahanshahi

***Abstract*—Regular inspection of nuclear power plant components is important to guarantee safe operations. However, current practice is time consuming, tedious, and subjective, which involves human technicians reviewing the inspection videos and identifying cracks on reactors. A few vision-based crack detection approaches have been developed for metallic surfaces, and they typically perform poorly when used for analyzing nuclear inspection videos. Detecting these cracks is a challenging task since they are tiny, and noisy patterns exist on the components' surfaces. This study proposes a deep learning framework, based on a convolutional neural network (CNN) and a Naïve Bayes data fusion scheme, called NB-CNN, to analyze individual video frames for crack detection while a novel data fusion scheme is proposed to aggregate the information extracted from each video frame to enhance the overall performance and robustness of the system. To this end, a CNN is proposed to detect crack patches in each video frame, while the proposed data fusion scheme maintains the spatiotemporal coherence of cracks in videos, and the Naïve Bayes decision making discards false positives effectively. The proposed framework achieves a 98.3% hit rate against 0.1 false positives per frame that is significantly higher than state-of-the-art approaches as presented in this paper.**

***Index Terms*—Crack detection, convolutional neural network (CNN), data fusion, deep learning, nuclear power plant inspection.**

I. Introduction

A. Motivation

THE U.S. is the world's largest supplier of commercial nuclear power. In 2015, 100 commercial reactors produced a total of 797 TW·h of electricity, accounting for 19.5% of the nation's total electric energy. Between 1952 and 2010, there have been 99 major nuclear power incidents worldwide that have cost more than $20 billion, where 56 of the incidents occurred in the U.S. [1]. One important factor for causing these incidents has been cracking that can lead to leaking. Nineteen of the above incidents were related to cracking or leaking that cost $2 billion. One important factor for causing incidents is cracking that may result in leaking. For instance, in 1996, a leaking valve caused an accident in the Millstone Nuclear Power Station in Waterford, CT, USA, which cost $254 million [1]. In 2010, leaked radioactive tritium from deteriorating underground pipes that cost $700 million at the Vermont Yankee Nuclear Power Plant in Vernon, VT, USA [1].

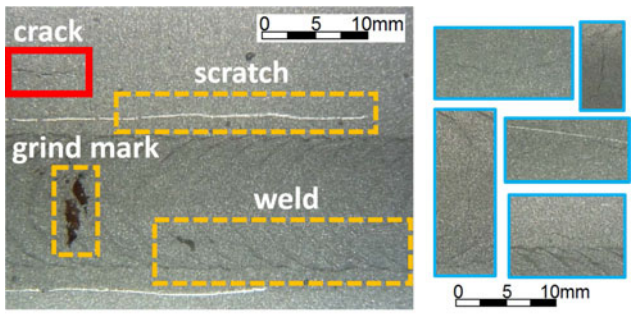

Fig. 1. Challenges of detecting cracks from inspection videos of nuclear power plants. (a) Crack with noisy patterns around it and (b) tiny cracks with low contrast and variant brightness.

Periodic inspection of reactors in nuclear power plants is crucial to ensure safe operations. Due to the hazardous environments aforementioned, a direct inspection is not feasible. Currently, many of the nuclear power plants conduct remote visual testing with radiation-hardened video systems [2] for inspecting reactors. A typical system includes a robotic arm that maneuvers a camera to remotely record videos of underwater component surfaces. Then, technicians review the videos and identify the cracks. This human-involved task is subjective, time consuming, and tedious.

The existence of tiny cracks and noisy patterns on metallic surfaces makes detecting cracks a very challenging task since most of the noisy patterns have linear shapes and stronger contrast compared to the tiny cracks. Fig. 1(a) shows a sample video frame with a crack and its surrounding noisy patterns that include a scratch, a grind mark, and a weld. Fig. 1(b) demonstrates samples of tiny cracks with low contrast and variant brightness that are hardly visible.

Manuscript received July 20, 2017; revised September 12, 2017; accepted October 3, 2017. Date of publication October 19, 2017; date of current version January 16, 2018. *(Corresponding author: Fu-Chen Chen).*

F.-C. Chen is with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: chen1623@purdue.edu).

M. R. Jahanshahi is with the Lyles School of Civil Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: jahansha@purdue.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIE.2017.2764844

Although several vision-based crack detection approaches are developed for concrete, rock, or pavement surfaces, only a few approaches consider crack detection on metallic surfaces. Recent vision-based steel surface inspection systems have been reviewed in [3] including crack detection algorithms. Although those algorithms might achieve more than 90% true positive rates (TPRs) in their applications, they fail to detect cracks on metallic surfaces in nuclear power plants [4].

The majority of existing approaches focus on detecting cracks in a single image. If a crack is not detected or a noisy pattern is falsely detected as a crack in the image, no other information is available to correct the detection results. Also, if a stitched image from video frames is used for crack detection, the stitching process might blur or even completely remove high frequency components (e.g., edges) by blending the overlapping regions of frames. This makes detecting tiny cracks much harder. Recently, an approach [4] was developed that fuses information obtained from multiple video frames and significantly improves the detection reliability. This study proposes a new approach based on Naïve Bayes with a convolutional neural network (CNN) that considers the spatiotemporal coherence of video frames and achieves even higher hit rates.

B. Related Works

Edge detection and morphological operations are popular approaches that extract local changes in image intensity for detecting cracks. They perform well for concrete or pavement surfaces while the cracks have stronger edges than noisy patterns [5]–[7]. For more complicated scenes, using advanced image analysis techniques, including image percolation [8], local binary pattern (LBP) [4], [9], crack blob features [10], Gabor [11], and wavelet [12], is a better strategy to detect cracks.

For metallic surfaces, vision-based crack detection methods [13], [14] have been reviewed in [3] for steel surface inspection during production. For detecting cracks in nuclear power plants inspection videos, Naïve Bayes classifier, principal component analysis, and anomaly detection were used in [15]. A method was developed in [4] based on LBP, the support vector machine (SVM), and data fusion using Bayes' theorem.

Recently, deep learning methods based on neural networks have dominated the speech recognition as well as vision-based pattern recognition areas [16]. For other areas including fault-tolerant control systems [17], for instance, using adaptive neural networks [18] also achieves successful results that compensates the actuator failures in nonlinear systems. In particular, CNNs have brought breakthroughs toward object detection and recognition [19]. The CNN requires a huge amount of annotated data for training, and many researchers have constructed large image datasets [20] for this purpose.

Several researches have been conducted for a variety of applications using the CNN. To detect objects in real time, region-based convolutional neural network (R-CNN) [21], Fast R-CNN [22], and Faster R-CNN [23] were developed. For object detection from videos, tubelets with convolutional neural network (T-CNN) was proposed in [24] based on R-CNN that won the object-detection-from-video (VID) task in the ImageNet Large-Scale Visual Recognition Challenge 2015 (ILSVRC 2015) [25]. Although R-CNN achieves real-time object detection, it has a limitation that the width–height ratio of rectangular region proposals cannot be too large or small. In nuclear power plant inspection videos, cracks are typically thin and long with variant shapes and orientations. Thus, R-CNN is not applicable since the region proposals of the cracks may violate the aforementioned limitations. Due to CNN's outstanding performance, several recent studies applied it for defect detection including railroad defects on steel surfaces [26], road cracks [27], concrete cracks [28], and cracks on nuclear power plant components [29].

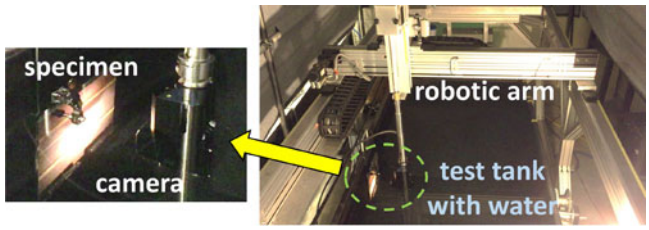

Fig. 2. Underwater camera system with a robotic arm scanner for video recording.

C. Contribution

A study [4] introduced a methodology that integrates LBP [30], SVM [31], and data fusion through Bayes' theorem to detect cracks in nuclear inspection videos (referred to LBP-SVM from hereafter). The LBP-SVM outperforms state-of-the-art crack detection algorithms that adopted undecimated wavelet transform (UWT) [13], morphological operations [7], and Gabor filtering [14].

This study proposes a new framework, called NB-CNN, based on a CNN and a Naïve Bayes data fusion scheme. The contribution of this study is threefolded.

1) A CNN architecture is proposed that detects crack patches more accurately than an LBP-SVM.
2) A registration procedure that maintains the spatiotemporal coherence of cracks in videos is proposed.
3) A Naïve Bayes data fusion scheme is proposed based on the statistics that discards false positives effectively through aggregating information from multiple frames.

The same framework can be applied to other defect detection applications where the CNN is trained using the corresponding datasets.

II. Data Collection

A. Inspection Videos

To develop and evaluate the proposed framework, videos of 20 underwater specimens that represented internal nuclear power plant components were collected. The specimens were made of 304 stainless steel with media blasting to limit glare from the camera lights. The widths and heights of the specimens were approximately 267 mm. Each specimen had weld crowns, different number of grinding marks, scratches, and cracks on the surface that are normally found on internal nuclear power plant components.

An underwater camera system commonly used in the field recorded the videos with 30 ft/s and 720 $\times$ 540 pixels resolution. The specimens were located inside a test tank filled with water where a robotic arm scanner maneuvered the camera (see

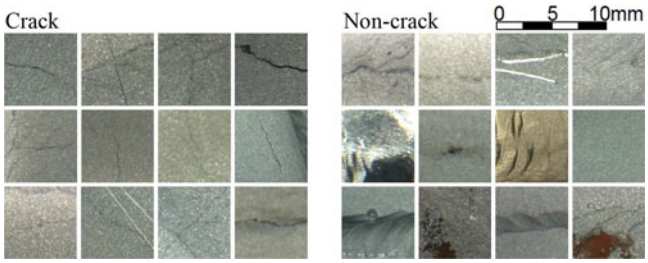

Fig. 3. Samples of 147 344 crack and 149 460 noncrack image patches.

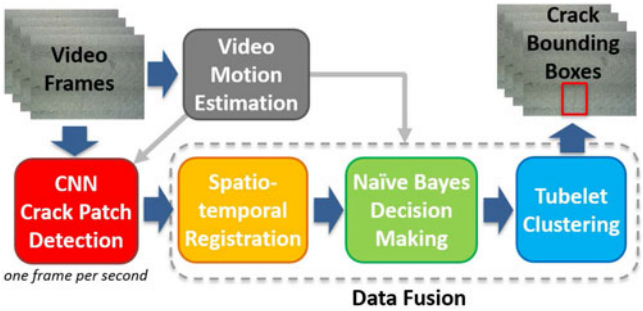

Fig. 4. Overview of the proposed NB-CNN framework.

Fig. 2). The dimensions of the scanner system were 122 cm × 152 cm × 305 cm, and it had four degrees of freedom (i.e., X, Y, Z, and rotation). The camera was placed approximately 10 cm from the specimen surface and moved slowly at a speed ranging from 1.27 to 3.85 mm per second during data collection. It took approximately 5–18 min to complete the scan for a specimen where the slow scanning speed minimized the occurrence of motion blur. During each recording, the camera's field of view remained constant. The image scales ranged from 56.1 to 74.3 μm per pixel and the crack widths varied from two to six pixels (i.e., 112.2–445.8 μm). The total length of collected videos was 199 min and 18 s (358 740 frames).

B. Image Patch Dataset

To train and validate the CNN, this study generated crack and noncrack image patches of 120 × 120 pixels from the video frames. Originally, 5326 crack image patches were manually annotated. Most of the cracks in the dataset were horizontal or vertical with at most ±15° slants. To detect cracks of different orientations and increase the variety of the dataset, the crack image patches were first rotated by 22.5°, 45°, and 67.5°, and then, flipped and rotated by 90°, 180°, and 270°. The pixel values of each image patch were also multiplied by a truncated Gaussian random variable, ranging from 0.85 to 1.20 with 1.00 mean and 0.08 standard deviation, to simulate brightness variations. Noncrack image patches were randomly cropped from background regions of the video frames. The final dataset contained 147 344 crack and 149 460 noncrack image patches. Fig. 3 illustrates samples of image patches.

III. Methodology

Fig. 4 demonstrates the overview of the proposed NB-CNN framework. Video motion estimation estimates the motion vector between successive frame pairs. CNN crack patch detection uses the proposed CNN to detect crack patches in each frame where one frame per second is analyzed. Data fusion aggregates the information obtained from multiple frames. It consists of three parts: spatiotemporal registration registers crack patches to a global spatiotemporal coordinates and forms crack tubelets, Naïve Bayes decision making determines whether a crack tubelet is a real crack or not, and tubelet clustering groups tubelets into crack clusters then generates crack bounding boxes.

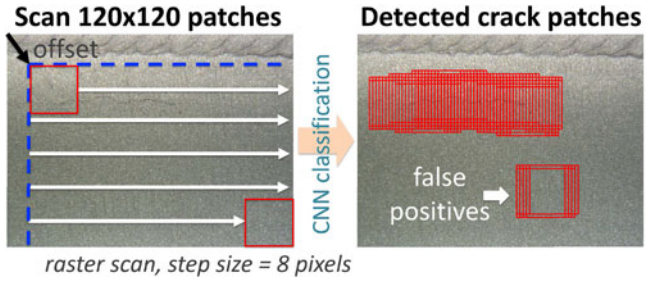

Fig. 5. CNN crack patch detection procedure: Scan of the frame with 120 × 120 patches (left) and detected crack patches using the CNN (right).

A. Video Motion Estimation

This procedure aims to estimate the frame movements for CNN crack patch detection and data fusion. During the recordings, the camera's field-of-view and the surface-camera distance remained constant. Thus, only translation occurred in the videos. As a result, this procedure applies a block-based motion estimation to compute motion vectors between successive frame pairs. Based on template matching, the motion vector MV_i is the displacement between a central inner block region within frame_i and its best match among a search range in frame_{i+1}. To this end, the sum of absolute difference (SAD) of pixel intensities is used as the matching criterion. Having all the motion vectors, the movement $\text{MOV}_{i,i+k}$ from frame_i to frame_{i+k} equals $\text{MV}_i + \text{MV}_{i+1} + \cdots + \text{MV}_{i+k-1}$, for $k > 0$. In this study, the inner block region has half the width and height of the video frame (i.e., 360 × 270 pixels). The search range is 10 pixels larger than the inner block region in width and height. Only one out of 16 pixels are sampled when calculating SAD to reduce computation cost.

B. CNN Crack Patch Detection

At this stage, each video frame is scanned with patches of size 120 × 120 pixels in raster scan order with step size of eight pixels. Then, the proposed CNN classifies each patch as a crack or noncrack patch. Each scanning has a starting 2-D offset ranging from $(0, 0)$ to $(7, 7)$ as illustrated in Fig. 5. The offset of frame_i equals $-\text{MOV}_{1,i}$ modulo eight (i.e., the step size). These offsets ensure the spatiotemporal consistency of patches that is discussed in more details in Section III-C.

The right side of Fig. 5 shows samples of detected crack patches where some of them are false positives. These false positives will be discarded by utilizing Naïve Bayes decision

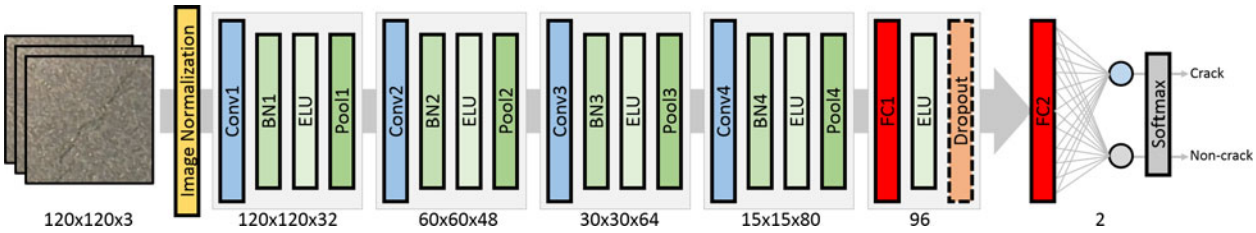

Fig. 6. Overall architecture of the proposed CNN. The numbers below layers indicate the output data size of each convolution or fully connected layer. Conv: Convolution layer; BN: Batch normalization layer; Pool: Pooling layer; ELU: Exponential linear unit layer; FC: fully connected layer.

TABLE I
CONVOLUTION, POOLING, AND FULLY CONNECTED LAYER CONFIGURATIONS OF THE PROPOSED CNN

Layer	Kernel Shape	Kernel #	Stride	Variables
Conv1	$11 \times 11 \times 3$	32	1	11 648
Pool1	$7 \times 7 \times 1$	–	2	–
Conv2	$11 \times 11 \times 32$	48	1	185 904
Pool2	$5 \times 5 \times 1$	–	2	–
Conv3	$7 \times 7 \times 48$	64	1	150 592
Pool3	$3 \times 3 \times 1$	–	2	–
Conv4	$5 \times 5 \times 64$	80	1	128 080
Pool4	$3 \times 3 \times 1$	–	2	–
FC1	5120	96	–	491 616
FC2	96	2	–	194

Fig. 7. Visualizations of 32 trained kernels in the first convolution layer of the proposed CNN.

making at a later stage. Detecting cracks in every frame is unnecessary since successive frames have significant overlap. So, one frame per second is analyzed in this study.

1) Overall Architecture of the Proposed CNN: Fig. 6 presents the overall architecture of the proposed CNN. The input is 3-D data: a 120×120 image patch of R, G, and B channels. The image normalization linearly scales each channel to have zero mean and unit L^2-norm. Then, the data go through different layers and a softmax layer predicts whether the input image patch is a crack or not. Table I lists the configurations of convolution, pooling, and fully connected layers. The proposed architecture follows the model used in TensorFlow [32] CNN tutorial with some modifications. More layers were added until the test error did not improve anymore, and the hyperparameters were fine tuned based on the guidelines described in [33].

A convolution layer performs a 3-D convolution with several kernels (i.e., filters) that serves as a finite impulse response filtering with a given stride (i.e., step size) to extract image edges and corners of different frequencies and orientations. Fig. 7 shows the visualizations of 32 trained kernels in the first convolution layer of the proposed CNN. Some kernels with black line segments work as edge detectors to extract features of cracks. Some kernels with irregular patterns perform as texture feature extractor that can help distinguish cracks from background. A pooling layer performs a 3-D maximum filtering with a given stride for applying a nonlinear transformation locally and data downsampling.

A Batch normalization [34] layer acts as a regularizer, enables higher learning rates to speed up training, and improves the performance of the CNN. It linearly transforms the data in each channel such that they have a distribution of zero mean and unit variance. An ELU serves as a nonlinear activation layer that outperformed any other activation function [35]. The first fully connected layer flattens the data and the second one serves as the final classifier. The softmax layer gives the final two scores (i.e., decision values) of being a crack and noncrack. The two scores range from zero to one and sum up to one. The CNN identifies the input as a crack patch if the score of being a crack (denoted as s^c) is greater than 0.5, and a noncrack patch otherwise. A dropout layer [36] disconnects some connections randomly during the training phase to prevent overfitting.

2) Training: To optimize the variables in the convolution, batch normalization, and fully connected layers, this study uses stochastic gradient descent [37] with a simple momentum. The training took place on an Exact deep learning Linux server with Ubuntu 14.04. The server included two Intel Xeon E5-2620 v4 CPUs, 256-GB DDR4 memories, and four NVIDIA Titan X Pascal GPUs. TensorFlow [32] was used to train the CNN in Python. The batch size was $n = 64$, the initial learning rate was $\tau = 0.002$, which decayed by 0.1 every 350 epochs, and the regularization weight was $\lambda = 0.0001$. One GPU was used for training where the training converged after 70 epochs (i.e., 32 535 s).

C. Spatiotemporal Registration

One major advantage of detecting objects in videos is that an object can be observed at different video frames (i.e., times). Analyzing more frames results in a more robust detection compared to processing only one frame. After obtaining the detection score s^c for each patch in different frames, patches of the same physical regions are registered based on their spatiotemporal coherence.

The concept of tubelets was introduced in [24] where the observations of the same object in different video frames were used in conjunction with a CNN to detect objects. This approach was called T-CNN. The locations of an object in different frames were estimated based on object tracking and optical flow. In this study, however, the patches are registered into a global spatiotemporal coordinate system where the

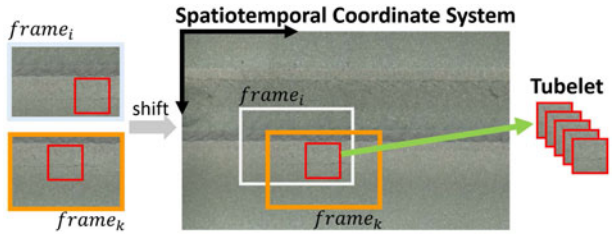

Fig. 8. Example of the spatiotemporal registration.

spatiotemporal coordinates represent the physical locations of patches on the surface that is under inspection. To this end, every patch in frame$_i$ is shifted by $-\text{MOV}_{1,i}$. All the shifted patches from different frames that have the same position in the spatiotemporal coordinate system correspond to the same region on the physical surface. Fig. 8 shows an example of the aforementioned registration. In this figure, both frame$_i$ and frame$_k$ include a corresponding crack patch. After registration, the shifted patches correspond to the same crack region on the physical surface in the spatiotemporal coordinate system as shown in Fig. 8. The scanning offsets introduced in Section III-B compensate the frame movements to align corresponding patches in the spatiotemporal coordinates that are obtained from different frames. Without the offsets, the corresponding patches do not cover the same exact regions of the physical object.

All the shifted patches that correspond to the same position in the spatiotemporal coordinate system form a tubelet if at least one of them is a detected crack patch (i.e., has the detection score of $s^c > 0.5$). In other words, a tubelet contains the observations (i.e., detection scores) of a physical location on the surface at different times in the video. During this process, missdetected crack patches may be included in tubelets, while false positives will form falsely detected tubelets. Such falsely detected tubelets are discarded through Naïve Bayes decision making as described in Section III-D.

D. Naïve Bayes Decision Making

To determine whether a tubelet is a crack or not, a general machine learning classifier that requires fixed-size input (e.g., SVM) is not applicable since each tubelet has different number of patches (i.e., observations). This study uses Bayes' theorem to provide a robust decision making. Assume a tubelet consists of n patches, and $P(C_{\text{crk}}|s_1^c, ..., s_n^c)$ and $P(C_{\text{ncrk}}|s_1^c, ..., s_n^c)$ represent the posterior probabilities of being a crack and noncrack, respectively. The decision making determines the tubelet as a crack if

$$\frac{P(C_{\text{crk}}|s_1^c, ..., s_n^c)}{P(C_{\text{ncrk}}|s_1^c, ..., s_n^c)} > \theta \tag{1}$$

where θ controls the sensitivity of the decision making, and s_i^c is the score obtained from the CNN for the ith patch. Since CNN computes s^c for each patch independently from other patches, a Naïve conditional independence assumption is used where $f(s_i^c|s_{i+1}^c, ..., s_n^c, C) = f(s_i^c|C)$, while $f(\cdot)$ is the probability density function (PDF). Rewriting the aforementioned equation

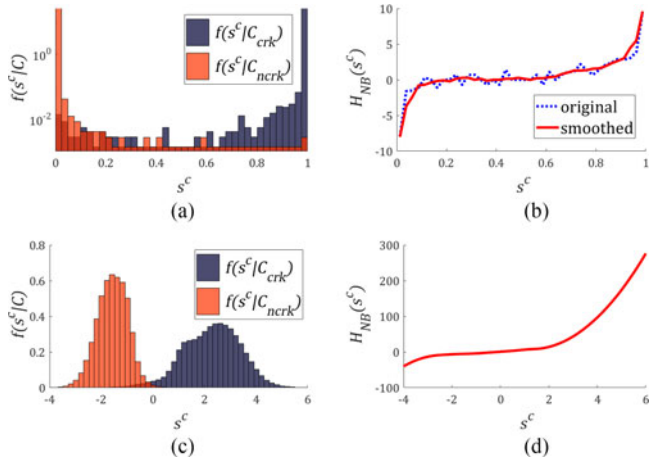

Fig. 9. (a) Likelihood functions for the proposed CNN, (b) $H_{\text{NB}}(\cdot)$ for the proposed CNN, (c) likelihood functions for the LBP-SVM, and (d) $H_{\text{NB}}(\cdot)$ for the LBP-SVM. The y-axis in (a) uses logarithmic scale.

and taking log on both sides, the equation becomes

$$\log \frac{P(C_{\text{crk}}) \prod_{i=1}^{n} f(s_i^c|C_{\text{crk}})}{P(C_{\text{ncrk}}) \prod_{i=1}^{n} f(s_i^c|C_{\text{ncrk}})} > \log \theta \tag{2}$$

or

$$\sum_{i=1}^{n} (\log f(s_i^c|C_{\text{crk}}) - \log f(s_i^c|C_{\text{ncrk}})) = \sum_{i=1}^{n} H_{\text{NB}}(s_i^c) > \theta_t \tag{3}$$

where $f(s_i^c|C_{\text{crk}})$ and $f(s_i^c|C_{\text{ncrk}})$ are likelihood functions, $H_{\text{NB}}(\cdot)$ converts the detection scores to a logarithmic likelihood ratio, and $\theta_t = \log \theta - \log P(C_{\text{crk}}) + \log P(C_{\text{ncrk}})$ controls the sensitivity. Estimating the prior probabilities $P(C_{\text{crk}})$ and $P(C_{\text{ncrk}})$ is not necessary since θ_t already contains them. By applying the CNN to validation data, the statistics for the likelihood functions are estimated. For a given tubelet, the summation of all likelihood ratios is computed. If the summation is greater than θ_t, the tubelet is classified as a crack; otherwise, the tubelet and the patches within it are discarded as being false positives (i.e., noncracks). The optimum value for θ_t is -28 in this study.

Fig. 9(a) and (b) shows the estimated likelihood functions and $H_{\text{NB}}(\cdot)$ of the proposed CNN in this study. The y-axis in Fig. 9(a) is in logarithmic scale since more than 98% of the samples lie on the first and last bars of the PDF. As shown in Fig. 9(b), the original $H_{\text{NB}}(\cdot)$ function contains fluctuations. So, the smoothed $H_{\text{NB}}(\cdot)$ is used in this study that is approximately an increasing function. The increasing characteristic of likelihood ratio matches the intuition that a higher detection score results in a larger likelihood ratio. Fig. 9(c) and (d) shows the likelihood functions and $H_{\text{NB}}(\cdot)$ for the LBP-SVM. The s^c of the LBP-SVM is the decision value of the SVM. For the LBP-SVM, a patch is classified as a crack patch if its $s^c > 0$; otherwise it is classified as a noncrack patch. Fig. 10(a) and (b) illustrates samples of crack patches before and after applying the proposed decision making scheme. Although several false positive patches exist in Fig. 10(a), this procedure discards them successfully [see Fig. 10(b)].

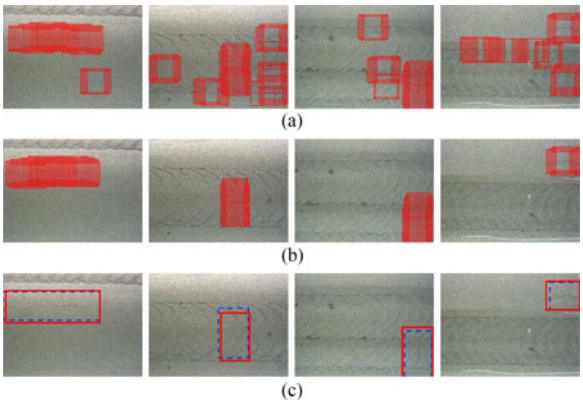

Fig. 10. (a) Sample crack patches including false positives before data fusion, (b) sample crack patches after Spatiotemporal Registration and Naïve Bayes Decision Making, and (c) sample crack bounding boxes (red line) and the ground-truth boxes (blue dashed line) after "Tubelet Clustering."

E. Tubelet Clustering

Each tubelet only presents a portion of a crack. To address this issue, nearby tubelets are grouped together as clusters after the false positive tubelets are discarded by Naïve Bayes decision making. This grouping is based on a hierarchical clustering approach that uses Euclidean distance as the grouping criterion with the cutoff equal to 20 pixels (i.e., if the Euclidean distance between two tubelets is less than 20 pixels, they are grouped together).

For each cluster, the likelihood ratios for all the tubelets within the cluster are added together. If this summation is greater than a threshold θ_c, the cluster is identified as a real crack; otherwise, the cluster is discarded as a false positive. In each frame, the smallest rectangle that contains all the patches corresponding to a detected crack cluster is used as the bounding box for that crack. θ_c controls the sensitivity of the overall detection. For instance, the detection achieves 98.3% hit rate against 0.1 false positives per frame when θ_c is 8.7 in this study.

The noncrack tubelets are discarded by Naïve Bayes decision making before tubelet clustering since tubelets have stronger spatiotemporal coherence than clusters. All the patches in a tubelet correspond to the same physical location in the spatiotemporal coordinate system. Thus, all of them should simultaneously be crack or noncrack patches. On the other hand, a noncrack tubelet might happen to be adjacent to a set of crack tubelets. Without discarding noncrack tubelets first, a cluster might be a mixture of crack and noncrack tubelets. This will affect the Naïve Bayes decision making and the shape of crack bounding boxes. As shown in Fig. 10, the proposed data fusion scheme successfully discards false positives and generates the bounding boxes of crack clusters that can truly represent the real cracks [see Fig. 10(c)].

IV. Experimental Results

In [4], the LBP-SVM outperformed state-of-the-art crack detection algorithms including UWT [13], morphological operations [7] referred to as Morph, and Gabor filtering [14] for crack detection. In this Section, the experimental results show that the proposed NB-CNN framework even performs better than the LBP-SVM. Furthermore, it is shown that the Naïve Bayes is more effective than other methods for discarding noncrack tubelets, and hence, it is more appropriate for data fusion when video frames are used.

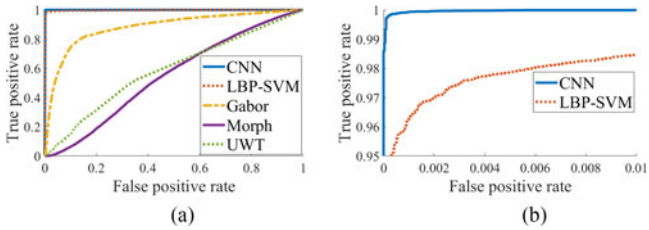

Fig. 11. (a) ROC curves of the proposed CNN, LBP-SVM, and three state-of-the-art approaches and (b) close view of ROC curves for the CNN and LBP-SVM.

A. Patch-Based Performance Evaluation

To evaluate the performance of the proposed CNN for crack patch detection, this study used all the 147 344 crack and 149 460 noncrack image patches described in Section II-B. Eighty percent of the image patches were used for training and 20% for validating the CNN and generating the receiver operating characteristic (ROC) curves. In the figures of ROC curves, the TPR is the number of true positives divided by the total number of positives, and the false positive rate (FPR) is the number of false positives divided by the total number of negatives. A classifier with low FPR (e.g., smaller than 1%) is desirable to detect crack patches without generating too many false positives in a given a frame.

Fig. 11(a) shows the ROC curves of the proposed CNN, LBP-SVM, Gabor, Morph, and UWT approaches. It indicates that the proposed CNN and LBP-SVM have much higher TPRs compared to the other three approaches. Although the CNN and LBP-SVM seem to have close ROC curves, Fig. 11(b) gives a close view of the curves. The same figure indicates that the CNN reaches high TPR faster than LBP-SVM. Thus, as shown in Section IV-B, the CNN achieves higher hit rates in frame-based performance evaluation. With 0.1% FPR, the CNN achieves 99.9% TPR while LBP-SVM has 96.2% TPR. This means when the CNN missdetects 0.1% of the crack patches, LBP-SVM may missdetect 3.8% of the crack patches that is 38 times more than the CNN's missdetection.

B. Frame-Based Performance Evaluation

Although the CNN has high TPR with very low FPR, it may still yield to false positive patches, as shown in Fig. 10(a). To address this issue, the proposed data fusion scheme maintains the spatiotemporal coherence of the patches and discards false positive tubelets based on Naïve Bayes decision making. This section shows how the hit rates are improved for the overall crack detection in videos when the data fusion is applied.

For evaluation purposes, 65 video segments (i.e., 41 370 frames) with cracks and 41 video segments (i.e., 45 180 frames) without cracks were used. In these video segments, the

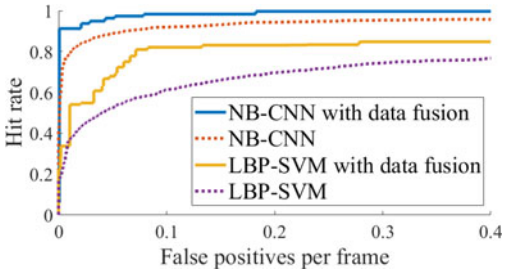

Fig. 12. Hit rate versus false positives per frame (FPPF) curves of the NB-CNN and LBP-SVM.

scanned area by the camera varied from 67.7 × 30.3 to 114.4 × 40.1 mm^2. In this study, one frame per second was processed that led to total 2885 frames for the evaluation process.

To obtain the ground truths of crack bounding boxes, first, the smallest bounding box for each crack was manually annotated. Since the proposed method obtains crack bounding boxes from patches of 120 × 120 pixels, these bounding boxes are slightly larger than the manually annotated ones. To conduct a fair evaluation, this study extended the annotated boxes by 120 pixels in width and height. The extended boxes served as the ground truths. To compute hit rates, the rules in the PASCAL object detection challenge [38] were used: A detected crack bounding box hits the ground truth if

$$\frac{\text{area}(B_d \cap B_{\text{gt}})}{\text{area}(B_d \cup B_{\text{gt}})} \geq 50\% \tag{4}$$

where B_d and B_{gt} are detected and ground truth bounding boxes, respectively.

Fig. 12 illustrates the hit rate versus false positives per frame (FPPF) curves of the NB-CNN and LBP-SVM. Without data fusion, each cluster is formed by spatially grouping the patches in a frame. In other words, the clustering is done within each frame independently from other frames without considering the spatiotemporal coherence. Fig. 12 shows that the NB-CNN outperforms an LBP-SVM. Additionally, this figure shows that the proposed data fusion scheme improves the hit rates for both NB-CNN and LBP-SVM. With 0.1 FPPF, the NB-CNN achieves 98.3% hit rate, whereas the LBP-SVM has only 82.1% hit rate when data fusion is used. As shown in Fig. 12, for 0.1% FPPF, NB-CNN's miss rate (i.e., 100% hit rate) is 1.7% and 8.0% with and without data fusion, respectively. These values are 17.9% and 38.7% for the LBP-SVM. The reason why an NB-CNN outperforms the LBP-SVM is that the CNN autonomously learns the optimum image features in convolution layers (e.g., Fig. 7) for crack detection while the LBP-SVM uses engineered features (i.e., features designed by human). Thus, the CNN has higher TPRs in Fig. 11(b) and results in higher hit rates for the NB-CNN (see Fig. 12).

To show the effectiveness of the Naïve Bayes decision making, this study compares four other methods for aggregating detection scores of patches within a tubelet. The first method intuitively uses the number of crack patches within a tubelet. The second method uses the sum of the detection scores within a tubelet. The third one is applied in T-CNN [24] that uses the top-k score (i.e., $s^c_{k\text{th}}$: the kth largest detection score within a tubelet). The forth one is adopted in an LBP-SVM [4] that sums up the likelihood ratios using a simpler model based on Bayes' theorem. Following is the mathematical representation of each approach where a tubelet is considered a real crack if

$$\text{No. of cracks: } \sum_{i=1}^{n} \{s_i^c > T\} > \theta_t \tag{5}$$

$$\text{Sum of scores: } \sum_{i=1}^{n} (s_i^c - T) > \theta_t \tag{6}$$

$$\text{Top-}k \text{ [24]} : s^c_{k\text{th}} > \theta_t \tag{7}$$

$$\text{Simple Bayes [4]} : \sum_{i=1}^{n} H_{\text{SB}}(s_i^c) > \theta_t \tag{8}$$

$$\text{Naïve Bayes: } \sum_{i=1}^{n} H_{\text{NB}}(s_i^c) > \theta_t \tag{9}$$

where n is the number of patches, $\{\cdot\}$ is an indicator function, T is the decision value threshold equal to 0.5 for the CNN and 0 for the LBP-SVM, and $H_{\text{SB}}(\cdot)$ is the likelihood ratio used in [4]. In sum of scores, T is subtracted from each score so that the summation penalizes the existence of noncrack patches in a tubelet. $H_{\text{SB}}(\cdot)$ is defined as following

$$H_{\text{SB}}(s^c) = \begin{cases} \log P(S^c > T|C_{\text{crk}}) - \log P(S^c > T|C_{\text{ncrk}}), & \text{if } s^c > T \\ \log P(S^c \leq T|C_{\text{crk}}) - \log P(S^c \leq T|C_{\text{ncrk}}), & \text{if } s^c \leq T \end{cases} \tag{10}$$

where S^c is the random variable corresponding to detection score. $P(S^c > T|C_{\text{crk}})$ and $P(S^c > T|C_{\text{ncrk}})$ are the TPR and FPR for the patch classification obtained from validation process, respectively. $P(S^c \leq T|C_{\text{crk}}) = 1 - P(S^c > T|C_{\text{crk}})$ and $P(S^c \leq T|C_{\text{ncrk}}) = 1 - P(S^c > T|C_{\text{ncrk}})$. That is, $H_{\text{SB}}(\cdot)$ in [4] is simply derived from TPR and FPR, whereas $H_{\text{NB}}(\cdot)$ in this study is derived based on the likelihood functions.

Table II lists the average areas under the hit rate versus FPPF curves (AUCs) of all the aforementioned methods and the optimized values of θ_t and k for quantitative comparison. This table shows that the Naïve Bayes attains the largest average AUC for both NB-CNN and LBP-SVM. As discussed in Section IV-A, an LBP-SVM may missdetect 38 times more crack patches than the CNN's detection. Too many crack patches are missdetected by the LBP-SVM, and not all of them can be restored by data fusion. Thus, regardless of the tubelet classification method, NB-CNN outperforms LBP-SVM with more than 17% of AUC.

In this study, CNN crack patch detection was implemented using TensorFlow [32] in Python and other procedures were implemented in MATLAB. Using the hardware system specified in Section III-B2, it took about 2.55 s to perform CNN crack patch detection on a 720 × 540 frame, while the other procedures only took 0.05 s. Although the computation time of the NB-CNN is a little bit longer than LBP-SVM's (i.e., 1.87 s), the NB-CNN provides better detections. Fig. 13 presents sample detection results using the proposed NB-CNN for different types

TABLE II
AVERAGE AUCS AND CORRESPONDING OPTIMIZED PARAMETERS OF DIFFERENT AGGREGATION METHODS FOR CLASSIFYING TUBELETS

Method		No. of Cracks	Sum of Scores	Top-k ($k = 4$) [24]	Simple Bayes [4]	Naïve Bayes
AUC	LBP-SVM	77.1%	79.0%	77.5%	79.0%	**79.2%**
	NB-CNN	95.5%	96.1%	96.3%	95.8%	**96.8%**
θ_t	LBP-SVM	4	1	0.2	13	11
	NB-CNN	5	−2	0.875	−38	−28

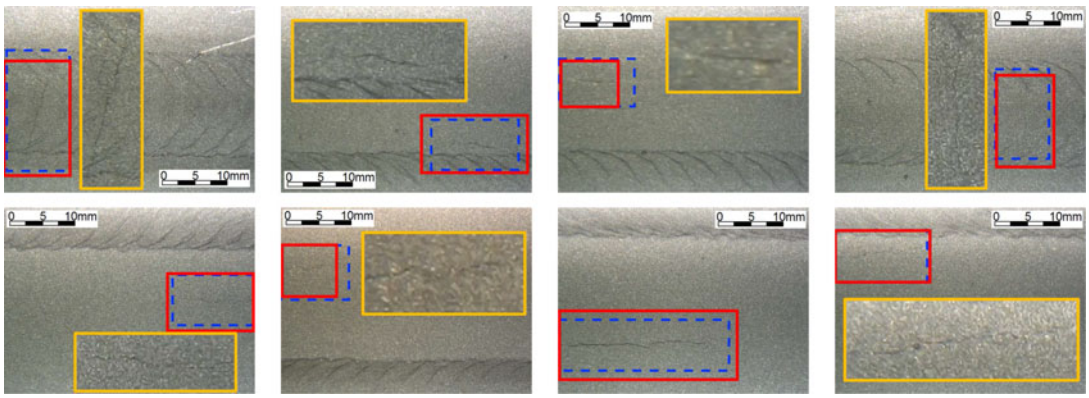

Fig. 13. Sample detection results from the proposed NB-CNN. The frames contain different types of cracks, backgrounds, and noisy patterns. Blue dashed: Ground truth; Red: Detected crack bounding boxes; Orange: Enlarged views of tiny cracks.

TABLE III
COMPUTATION TIMES FOR A 720 × 540 FRAME AND THE AVERAGE AUCS OF DIFFERENT SCANNING STEP SIZES IN "CNN CRACK PATCH DETECTION"

Step Size	4	6	8	12	16	20
Time (s)	9.35	4.10	2.55	1.16	0.69	0.43
AUC	96.3%	96.2%	**96.8%**	96.0%	95.1%	94.7%

of cracks, noisy patterns, and backgrounds. As seem from this figure, although some cracks are very tiny with low contrast and backgrounds are complex, the NB-CNN still leads to robust detections.

The value of patch scanning step size in CNN crack patch detection (see Section III-B) affects the density of patches in a frame and the computation time. Table III lists the computation times for a 720 × 540 frame and the average AUCs of different scanning step sizes. For step sizes four to eight, the corresponding AUC values are very close. The AUC starts to decrease when the step size is over 12. This study chooses step size of eight since it has the highest AUC and a reasonable computation time.

V. CONCLUSION AND DISCUSSION

Regular inspection of nuclear power plant components is an important task where current manual practice is time consuming and subjective. On the metallic surfaces of nuclear power plant reactors, the existence of tiny cracks and noisy patterns makes it very challenging to detect cracks autonomously. Only a few vision-based crack detection methods have been developed for metallic surfaces, and they fail to perform well for this application [4]. This study proposed a new deep learning framework called NB-CNN to detect cracks on underwater metallic surfaces from nuclear inspection videos. For detecting crack patches in each frame, the proposed CNN leads to 99.9% TPR against 0.1% FPR. Different from other approaches that focus on detecting cracks in a single image, a novel data fusion scheme is proposed that aggregates information obtained from multiple video frames and significantly improves the hit rates. In the proposed data fusion scheme, the spatiotemporal registration registers crack patches to a global coordinate system and forms tubelets, and the Naïve Bayes decision making discards noncrack tubelets more effectively than other methods. The proposed NB-CNN achieves 98.3% hit rate against 0.1 FPPF that is much higher than the 82.1% hit rate of LBP-SVM [4] with the same FPPF. The capability of the proposed NB-CNN is a significant achievement toward autonomous inspection of nuclear power plant internal components.

The main advantage of the proposed NB-CNN is that it achieves higher hit rate than other approaches with fast operation speed. The proposed NB-CNN can even detect tiny cracks with low contrast and variant brightness that are hardly visible. On the other hand, one disadvantage is that the CNN needs lots of training data (e.g., more than 100 000 samples) to make the training converge and prevent overfitting. Another disadvantage is that the computations of the CNN heavily rely on a GPU. Without using a GPU, the computations of the CNN might be ten times slower. Also, the proposed NB-CNN only detects crack locations without quantifying properties of cracks. As part of future works, the CNN has the capability to perform regression and image segmentation that can be used to quantify crack lengths and widths.

ACKNOWLEDGMENT

The authors would like to thank Prof. E. Delp from the School of Electrical and Computer Engineering, Purdue University, for his constructive feedback on this study.

REFERENCES

[1] B. K. Sovacool, "A critical evaluation of nuclear power and renewable electricity in asia," *J. Contemporary Asia*, vol. 40, no. 3, pp. 369–400, Aug. 2010.
[2] S. E. Cumblidge, M. T. Anderson, S. R. Doctor, F. A. Simonen, and A. J. Elliot, " An assessment of remote visual methods to detect cracking in reactor components," Pacific Northwest National Lab., Richland, WA, USA, Tech. Rep. PNNL-SA–57384, 2008.
[3] N. Neogi, D. K. Mohanta, and P. K. Dutta, "Review of vision-based steel surface inspection systems," *EURASIP J. Image Video Process.*, vol. 2014, no. 1, pp. 1–19, 2014.
[4] F.-C. Chen, M. R. Jahanshahi, R.-T. Wu, and C. Joffe, "A texture-based video processing methodology using Bayesian data fusion for autonomous crack detection on metallic surfaces," *Comput.-Aided Civil Infrastructure Eng.*, vol. 32, no. 4, pp. 271–287, Apr. 2017.
[5] I. Abdel-Qader, O. Abudayyeh, and M. E. Kelly, "Analysis of edge-detection techniques for crack identification in bridges," *J. Comput. Civil Eng.*, vol. 17, no. 4, pp. 255–263, Oct. 2003.
[6] Y. Fujita and Y. Hamamoto, "A robust automatic crack detection method from noisy concrete surfaces," *Mach. Vis. Appl.*, vol. 22, no. 2, pp. 245–254, Feb. 2010.
[7] M. R. Jahanshahi and S. F. Masri, "Adaptive vision-based crack detection using 3D scene reconstruction for condition assessment of structures," *Autom. Construction*, vol. 22, pp. 567–576, Mar. 2012.
[8] T. Yamaguchi and S. Hashimoto, "Fast crack detection method for large-size concrete surface images using percolation-based image processing," *Mach. Vis. Appl.*, vol. 21, no. 5, pp. 797–809, Feb. 2009.
[9] Y. Hu and C.-X. Zhao, "A novel LBP based methods for pavement crack detection," *J. Pattern Recog. Res.*, vol. 5, no. 1, pp. 140–147, 2010.
[10] M. R. Jahanshahi, S. F. Masri, C. W. Padgett, and G. S. Sukhatme, "An innovative methodology for detection and quantification of cracks through incorporation of depth perception," *Mach. Vis. Appl.*, vol. 24, no. 2, pp. 227–241, Dec. 2011.
[11] E. Zalama, J. Gómez-García-Bermejo, R. Medina, and J. Llamas, "Road crack detection using visual features extracted by gabor filters," *Comput.-Aided Civil Infrastructure Eng.*, vol. 29, no. 5, pp. 342–358, May 2014.
[12] G. P. Bu, S. Chanda, H. Guan, J. Jo, M. Blumenstein, and Y. C. Loo, "Crack detection using a texture analysis-based technique for visual bridge inspection," *Electron. J. Structural Eng.*, vol. 14, no. 1, pp. 41–48, Jan. 2015.
[13] X.-Y. Wu, K. Xu, and J.-W. Xu, "Application of undecimated wavelet transform to surface defect detection of hot rolled steel plates," in *Proc. Congr. Image Signal Process.*, May 2008, vol. 4, pp. 528–532.
[14] D.-C. Choi, Y.-J. Jeon, S. J. Lee, J. P. Yun, and S. W. Kim, "Algorithm for detecting seam cracks in steel plates using a Gabor filter combination method," *Appl. Opt.*, vol. 53, no. 22, pp. 4865–4872, Aug. 2014.
[15] S. J. Schmugge *et al.*, "Automatic detection of cracks during power plant inspection," in *Proc. 3rd Int. Conf. Appl. Robot. Power Ind.*, Oct. 2014, pp. 1–5.
[16] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.
[17] S. Yin, H. Luo, and S. X. Ding, "Real-time implementation of fault-tolerant control systems with performance optimization," *IEEE Trans. Ind. Electron.*, vol. 61, no. 5, pp. 2402–2411, May 2014.
[18] S. Yin, H. Yang, H. Gao, J. Qiu, and O. Kaynak, "An adaptive NN-based approach for fault-tolerant control of nonlinear time-varying delay systems with unmodeled dynamics," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 8, pp. 1902–1913, Aug. 2017.
[19] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 1–9.
[20] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 248–255.
[21] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 580–587.
[22] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1440–1448.
[23] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.
[24] K. Kang *et al.*, "T-CNN: Tubelets with convolutional neural networks for object detection from videos," *IEEE Trans. Circuits Syst. Video Technol.*, to be published, 2016.
[25] O. Russakovsky *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
[26] D. Soukup and R. Huber-Mörk, "Convolutional neural networks for steel surface defect detection from photometric stereo images," in *Proc. Int. Symp. Visual Comput.*, Dec. 2014, pp. 668–677.
[27] L. Zhang, F. Yang, Y. D. Zhang, and Y. J. Zhu, "Road crack detection using deep convolutional neural network," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2016, pp. 3708–3712.
[28] Y.-J. Cha, W. Choi, and O. Büyüköztürk, "Deep learning-based crack damage detection using convolutional neural networks," *Comput.-Aided Civil Infrastructure Eng.*, vol. 32, no. 5, pp. 361–378, Mar. 2017.
[29] S. J. Schmugge *et al.*, "Detection of cracks in nuclear power plant using spatial-temporal grouping of local patches," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Mar. 2016, pp. 1–7.
[30] R. Zabih and J. Woodfill, "Non-parametric local transforms for computing visual correspondence," in *Proc. 3rd Eur. Conf. Comput. Vis.*, May 1994, pp. 151–158.
[31] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995.
[32] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," arXiv:1603.04467 [cs.DC], Mar. 2016.
[33] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," arXiv:1206.5533 [cs.LG], Sep. 2012.
[34] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 448–456.
[35] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUs)," arXiv:1511.07289 [cs.LG], Feb. 2016.
[36] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014.
[37] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. 19th Int. Conf. Comput. Stat.*, 2010, pp. 177–186.
[38] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Sep. 2009.

Fu-Chen Chen received the B.S. degree in electrical engineering and the M.S. degree in computer science from National Taiwan University, Taipei, Taiwan, in 2008 and 2010, respectively. He is currently working toward the Ph.D. degree in electrical and computer engineering with Purdue University, West Lafayette, IN, USA.

His M.S. research topic was video processing and video analysis. He is currently a Research Assistant with Purdue University. Before studying at Purdue, he was a Senior Software Engineer in the Computer Vision Algorithm Development Team, Pixart Imaging Inc., Hsinchu, Taiwan (2010–2014). His research interests include computer vision, pattern recognition, machine learning, and image processing.

Mohammad R. Jahanshahi received the B.S. degree in civil engineering from Shiraz University, Shiraz, Iran, in 2001; the M.S. degree in structural engineering from Tarbiat Modarres University, Tehran, Iran, in 2004; and the M.S. degree in electrical engineering in 2006 and the Ph.D. degree in civil engineering in 2011, both from the University of Southern California, Los Angeles, CA, USA.

He is currently an Assistant Professor with the Lyles School of Civil Engineering, Purdue University, West Lafayette, IN, USA. Before joining Purdue, he was a Research Technologist in the Mobility and Robotic Systems Section, NASAs Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, USA. His research interests include autonomous sensing, data interpretation, and intelligent condition assessment of structures. He has been working in the field of computer vision and machine learning to develop robust systems for health monitoring of civil infrastructures.