DOI: 10.1111/mice.12533

ORIGINAL ARTICLE

COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING WILEY

Pavement defect detection with fully convolutional network and an uncertainty framework

Zheng Tong[1] | **Dongdong Yuan**[2] | **Jie Gao**[3] | **Zhenjun Wang**[4]

[1]Sorbonne Université, Université de Technologie de Compiègne, CNRS, Compiègne cedex, France

[2]School of Highway, Chang'an University, Xi'an, China

[3]School of Transportation and Logistics, East China Jiaotong University, Nanchang, China

[4]School of Materials Science and Engineering, Chang'an University, Xi'an, China

Correspondence
Zhenjun Wang, School of Materials Science and Engineering, Chang'an University, Xi'an 710061, China.
Email: zjwang@chd.edu.cn

Funding information
National Natural Science Foundation of China, Grant/Award Number: 51978067; Science and Technology Development Project of Xinjiang Production and Construction Corps, Grant/Award Number: 2019AB013; Key Research and Development Program of Shaanxi Province of China, Grant/Award Number: 2019GY-174; China Scholarship Council, Grant/Award Number: CSC201801810108

Abstract

Image segmentation has been implemented for pavement defect detection, from which types, locations, and geometric information can be obtained. In this study, an integration of a fully convolutional network with a Gaussian-conditional random field (G-CRF), an uncertainty framework, and probability-based rejection is proposed for detecting pavement defects. First, a fully convolutional network is designed to generate preliminary segmentation results, and a G-CRF is used to refine the segmentation. Second, epistemic and aleatory uncertainties in the model and database are considered to overcome the disadvantages of traditional deep-learning methods. Last, probability-based rejection is conducted to remove unreasonable segmentations. The proposed method is evaluated on a data set of images that were obtained from 16 highways. The proposed integration segments pavement distresses from digital images with desirable performance. It also provides a satisfactory means to improve the accuracy and generalization performance of pavement defect detection without introducing a delay into the segmentation process.

1 | INTRODUCTION

Digital imaging technology has become popular for autonomous pavement defect detection (Tong, Gao, Han, & Wang, 2018; K.C.P. Wang & Smadi, 2011). In the past 10 years, many algorithms have been utilized for automated defect detection using digital images, such as edge detectors (Abdel-Qader, Abudayyeh, & Kelly, 2003), cracktrees (Zou, Cao, Li, Mao, & Wang, 2012), and percolation models (Yamaguchi & Hashimoto, 2010). Also, a promising and new definition of crack width was proposed for its unambiguous measurement (N. Wang, Zhao, Li, Zhao, & Zhao, 2018). However, these unsupervised algorithms are limited by their unacceptable generalization performance and low stability in various real-world conditions (Cha, Choi, & Büyüköztürk, 2017).

Machine learning algorithms have improved the autonomous pavement crack detection significantly in the past decade. Artificial neural networks (ANNs) (Adeli & Samant, 2000; Rafiei, Khushefati, Demirboga, & Adeli, 2017), support vector machines (SVMs) (Lin & Liu, 2010), and restricted Boltzmann machine (Rafiei & Adeli, 2017, 2018) are typical machine-learning algorithms and have widely been used for defect detection in practice.

Nevertheless, these algorithms require a set of specific features to be provided to the algorithms, thereby images with different new features will not be properly analyzed, which results in inadequate generalization performance (Daniel & Preeja, 2014; Nejad & Zakeri, 2011).

Substantial developments in deep learning have provided a feasible approach for autonomous pavement defect detection (Cha et al., 2017; Zhang et al., 2017). The most promising advantage of deep learning is that objects are detected by learning features in small local receptive fields and fusing them into global features from example data autonomously (LeCun, Bengio, & Hinton, 2015; P. Wang & Bai, 2018). It provides a method for obtaining the type, location, and geometric information of objects such as defects with no human assistance (Molina-Cabello, Luque-Baena, López-Rubio, & Thurnhofer-Hemsi, 2018; Tong, Gao, Sha, Hu, & Li, 2018).

1.1 | Related works

Generally, the methods for pavement crack detection using deep learning can be classified into three groups: recognition-based, location-based, and segmentation-based methods. The recognition-based methods are the primal approach for detecting pavement cracks in digital pavement surface images. First, original pavement images are cropped into small pieces and imported into convolutional neural networks (CNNs) for distress recognition (Cha et al., 2017; Tong, Gao, & Zhang, 2018). Then, pavement defects are recognized and located in the cropped images. AlexNet (Gao, & Mosalam, 2018; Krizhevsky, Sutskever, & Hinton, 2012) and LeNet-5 (El-Sawy, Hazem, & Loey, 2016; Guo, Yao, & Chen, 2016) as typical CNNs are adopted and modified to detect defects. However, there are two main disadvantages of recognition-based methods: (a) the location results are coarse and depend on the sizes of the cropped images and (b) the geometric information cannot be obtained.

The methods of the second type for detecting pavement defects using deep learning are the location-based methods. In the second methods, deep-learning models are designed to locate a distress with a box. Under the location-based methods, region-based CNNs (R-CNNs) (Farfade, Saberian, & Li, 2015; Ren, He, Girshick, & Sun, 2015) and their modified models (Girshick, 2015; Girshick, Donahue, Darrell, & Malik, 2016) have been developed to overcome the disadvantages of the recognition-based methods. For example, faster R-CNNs are used for structural visual inspection (Cha, Choi, Suh, Mahmoudkhani, & Büyüköztürk, 2018), material property evaluation (Liu, Sha, Tong, & Gao, 2018), and crack detection (Kim, Ahn, Shin, & Sim, 2019). The location errors are reduced; however, the geometric information still cannot be obtained.

The segmentation-based methods, as the third type of method, acquire the geometric information of pavement defects by pixelwise classification. Pixelwise classification means a deep-learning algorithm classifies each pixel in a pavement image into one of the class memberships (e.g., pavement, crack, and pothole). Fully convolutional networks (FCNs) (Long, Shelhamer, & Darrell, 2015; L. Wang, Ouyang, Wang, & Lu, 2015) and their modified models (Shen, Gan, Yan, & Zeng, 2017; Zhou, Zhang, Lei, Li, & Tu, 2016) have achieved remarkable success in the field of image segmentation. For example, Zhang et al. (2017) proposed an automated pixel-level asphalt pavement crack detection approach using an FCN-based method. In their study, the algorithm has high precision and F-measure for crack detection. A CNN with a dense conditional random field (M. Wang & Cheng, 2019) is used to detect pipe defects. The results demonstrate that the model improves the defect segmentation precision obviously. Autonomous concrete crack detection (Dung & Anh, 2019) is also realized using deep FCNs. FCN-based methods outperform other methods in geometric information extraction and realize type recognition, defect location, and geometric information segmentation based on pavement digital images. In addition, FCNs are also used to detect other structure defects (L. C. Chen, Papandreou, Kokkinos, Murphy, & Yuille, 2017; N. Wang et al., 2018; Xue & Li, 2018). Therefore, it is feasible to use FCNs to detect pavement defects (e.g., cracks, potholes, and bleeding areas).

However, two problems remain: First, the defects, noise patterns, and transition regions are nearly indistinguishable in some digital images. It leads to the random errors in the defect labels, which can be regarded as data corruption (Sui, Feissel, & Denoeux, 2018) or aleatory uncertainty (Tran, Jauberthie, Le Gall, & Travé-Massuyès, 2018). For example, a random error occurs when an operator labels a small area of a crack as an area of pavement. Similar random errors are inevitable and always lead to unreasonable convergence. The probability of these random errors is not easy to be measured. It is considered as an aleatory uncertainty. Second, the information loss that is due to the translation variance (Krizhevsky et al., 2012) and the unreasonable structures of FCNs leads to the epistemic errors in the feature maps. For example, an algorithm in a crack detection system extracts a feature related to cracks ambiguously. The errors of these confusing features are considered as an epistemic uncertainty. If left unaddressed, a trained FCN model might be biased or even output ridiculous results.

The uncertainty framework and probability-based rejection can be used to overcome the problems that are due to the aleatory and epistemic uncertainty. The uncertainty framework is a technique for modeling uncertainty in optimization problems without using a probability distribution (Bertsimas, Brown, & Caramanis, 2011; Bertsimas, Dunn, Pawlowski, & Zhuo, 2018). Further, the uncertainty framework in the study is to represent an uncertainty problem as one or more parameters. It was proposed by Bi and Zhang

(2005) and Bhattacharyya, Pannagadatta, and Smola (2005). In practice, the uncertainty framework has been demonstrated to lead to improvements in many statistical problems. For example, Pant, Trafalis, and Barker (2011) proved that the uncertainty theory was able to deal with imbalanced data. S. Chen, Leng, and Labi (2019) developed an algorithm for simulating autonomous driving by considering prior knowledge. Prior knowledge can also be considered as an uncertainty. Studies have shown that uncertainty theory can be used to handle uncertainties in labels for machine learning (Biggio, Nelson, & Laskov, 2011; Natarajan, Dhillon, Ravikumar, & Tewari, 2013). Ben-Tal, Bhadra, Bhattacharyya, and Nemirovski (2012) showed that the uncertainty theory could be used to reduce feature uncertainty. Probability-based rejection is a method for distinguishing false results from reasonable results based on Bayes' rule (McGrayne, 2011). It has been widely used to remove false decisions (Laukkanen, 2016) and classification results (Moskát & Hauber, 2007) and has demonstrated success in handling aleatory and epistemic uncertainties. In summary, according to the literature review, the uncertainty framework and probability-based rejection have the potential to solve the uncertainty problems that are discussed above and to further improve the accuracy of FCNs. However, few studies have been conducted on integrating FCN, the uncertainty framework, and probability-based rejection.

1.2 | Contributions

This paper presents a combination of FCN with Gaussian-conditional random field (FCN-GCRF), the uncertainty framework, and probability-based rejection for pavement defect detection using digital images. The objective of this study is to develop a novel computer vision technique for further improving the precision of pavement defect detection, via which types, locations, and geometric information can be obtained. In the proposed technique, an FCN is used as a front end to preliminarily highlight the areas of pavement defects at the pixel level. A G-CRF and SVMs are combined with the FCN to refine these areas to improve the precision. Epistemic and aleatory uncertainties are integrated into the combination to overcome the disadvantages of the traditional deep-learning methods. Last, probability-based rejection is conducted to remove false segmentation. The proposed combination is still an end-to-end supervised algorithm. Our contributions in this study can be summarized as follows:

1. An FCN-based method that is deeply integrated with a G-CRF and SVMs for detecting pavement defects at the pixel level with satisfactory precision is proposed, via which types, locations, and geometric information can be obtained.
2. An uncertainty-framework method is presented for considering the aleatory and epistemic uncertainty (e.g., the errors in the features and the labels) to further increase the precision of pavement defect detection using deep-learning methods.
3. A probability-based method is presented for rejecting falsely identified areas of pavement defects in the segmentation results to increase the reliability of the output. Additionally, it demonstrates that there are always transition regions between defective areas and undefective areas on the asphalt pavement.

The remainder of the paper is organized as follows: In Section 2, the proposed combination is described in detail, including the structure of the FCN-GCRF, the procedures for integrating the uncertainty framework, and the probability-based rule for rejecting false segmentation results. In Section 3, the database and implementation details for the proposed combination are presented, followed by the results and a discussion in Section 4. The conclusions of this study are presented in Section 5.

2 | METHODOLOGY

The combination of the FCN with a G-CRF is described in Section 2.1. Then, the method for integrating the uncertainty framework with the FCN is presented in Section 2.2. The probability-based rejection rule is presented in Section 2.3. Last, the end-to-end combination model is summarized in Section 2.4.

2.1 | FCN with G-CRF

This section describes the proposed FCN with Gaussian-conditional random field and SVMs. The brief overview of the existing FCN is described in Section 2.1.1. In Section 2.1.2, a Gaussian-conditional random field is integrated with FCN to generate more reasonable dense feature maps. In Section 2.1.3, a method for combining SVMs and FCN to improve the generalization is presented.

2.1.1 | Brief overview of FCN

FCN is the state-of-the-art technology for image segmentation. An FCN for generating dense feature maps for pavement defect detection is designed, as shown in Figure 1. According to Figure 1, the dense feature maps are generated via convolution, pooling, and dilated convolution. The processes of convolution and pooling are expressed in Equations (1) and (2),

$$z^{l}_{\mu,\nu} = \sum_{i=-\infty}^{+\infty} \sum_{j=-\infty}^{+\infty} x^{l-1}_{\mu,\nu} \cdot k^{l}_{rot_{i,j}} \cdot \chi(i,j) + b^{l} \tag{1a}$$

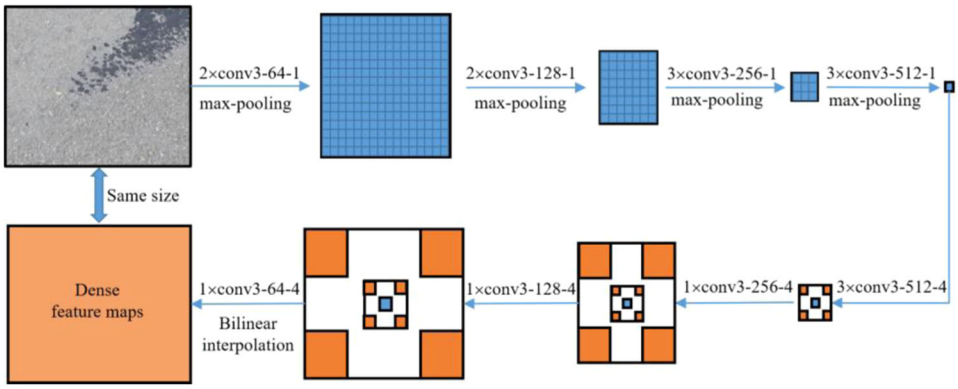
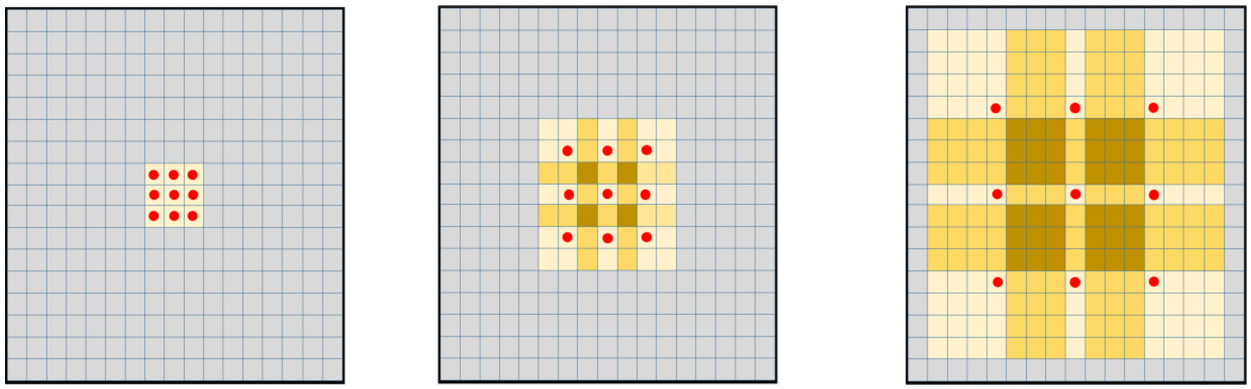


FIGURE 1 Architecture of FCN for generating dense feature maps



(a) A 1-dilated convolution (b) A 2-dilated convolution (c) A 4-dilated convolution

FIGURE 2 Examples of dilated convolution operations

$$\chi(i,j) = \begin{cases} 1, 0 \le i,j \le n \\ 0, \text{ other} \end{cases} \tag{1b}$$

$$x^{l}_{\mu,\nu} = \frac{1}{1+e^{z^{l-1}_{\mu,\nu}}} \tag{1c}$$

$$z^{l+1}_{\mu,\nu} = \max\left(x^{l}_{\mu,\nu}\right) + b^{l+1} \tag{2a}$$

$$x^{l+1}_{\mu,\nu} = \frac{1}{1+e^{z^{l+1}_{\mu,\nu}}} \tag{2b}$$

where $x^{l}{}_{\mu,\nu}$ and $x^{l\text{-}1}{}_{\mu,\nu}$ are the output and input of a kernel $k^{l}{}_{i,j}$ in the μth row and νth column from the lth convolutional layer; $k^{l}{}_{roti,j}$ is the transposed matrix of a kernel $k^{l}{}_{i,j}$; and b^{l} is the bias in the lth convolutional layer.

Previous studies (Shi, Ye, & Wu, 2016; Sun, Song, Jiang, Pan, & Pang, 2017) show that the pooling operation always leads to large spatial information loss in the processing of upsampling. Thus, dilated convolution is applied in the FCN for upsampling without increasing the number of parameters. A dilated convolution operation is considered as a normal convolution operation with a specified number of holes (zeros). Examples of dilated convolution operations are presented in Figure 2.

After the dilated convolution, the generated dense feature maps are upsampled via bilinear interpolation. By the process, the final dense feature maps have the same size as the input image. Each pixel in the feature maps as a pattern is classified into one of the class memberships by predicting a probability score vector. The pixel-level segmentation is achieved by predicting the class of each pixel.

2.1.2 | Combination of FCN and G-CRF

In an FCN, the dense feature maps are used to generate segmentation results at the pixel level. In a study (Chandra & Kokkinos, 2016), a G-CRF was introduced in the FCN for the pixel-level classification, as illustrated in Figure 3. As shown in Figure 3, the energy of dense feature maps is generated by combining the vertical and horizontal factors. Consider the output of the decoder in Figure 4 as a matrix that contains P pixels, where P is equal to the number of pixels in the imported image. Each pixel $p \in \{p_1,\ldots, p_P\}$ belongs to a class $l \in \{1,\ldots, L\}$. In the segmentation task, we specify the

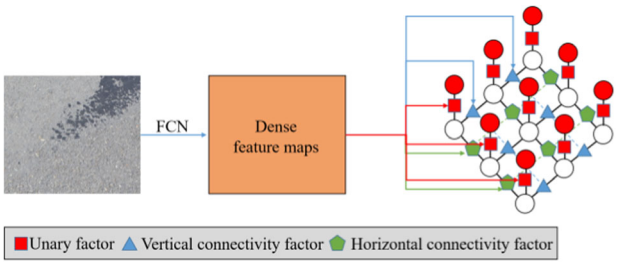


FIGURE 3 G-CRF for the pixel-level classification

pixel-level ground-truth labels $\mathbf{y} \in \mathbb{R}^P$ and an inferred hypothesis $\mathbf{x} \in \mathbb{R}^N$, where $N = P \times L$. A G-CRF defines the energy of the inferred hypothesis $\mathbf{x}$ as Equation (3):

$$E(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T (A + \lambda \mathbf{I})\,\mathbf{x} - B\mathbf{x} \tag{3}$$

where A and B are a symmetric $N \times N$ matrix of pairwise terms and an $N \times 1$ vector of unary terms, respectively. Therefore, $E(\mathbf{x})$ can be regarded as the posterior log-likelihood of the inferred hypothesis $\mathbf{x}$, which represents the vertical and horizontal connectivity relationships between pixels in the outputs of the decoder. Thus, it is more reasonable to use $E(\mathbf{x})$ for the pixel-level classification than the outputs of the decoder.

2.1.3 | Integration with support vector machines

In a traditional FCN integrated with a G-CRF (Chandra & Kokkinos, 2016), $E(\mathbf{x})$ is imported into fully connected layers and a softmax layer for pixel-level classification. Considering the generalization performance limits and the overfitting of fully connected layers (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014), SVMs are used to replace the structure in this study. As an SVM outperforms multiclassification in binary classification in practice (Jarray, Boughorbel, Mansour, & Tlig, 2018; R. Wang, Li, Li, & Zhang, 2019), the multiclassification problem is converted to multiple binary-classification problems, as shown in Figure 4. The classes of l_{SVM1}, l_{SVM2}, and l_{SVM3} are {Pavement, Defect}, {Crack}, and {Pothole, Bleeding}.

As the uncertainty framework is integrated with the SVMs in Section 2.2, the foundation of SVM is briefly recalled. For the remainder of the paper, let $\{\boldsymbol{x}_i, y_i\}^n_{i=1}$ be the training set that is provided for the pixel-level classification in SVMs, where $\boldsymbol{x}_i \in \mathbb{R}^p$ and y_i is the label for the task. Namely, $\boldsymbol{x}_i$ ($i = 1, \ldots, n$) denotes the energies of the inferred hypothesis that are generated by the G-CRF.

An SVM, which is a variation on the simpler maximal margin classifier, uses the hinge loss as the loss function and balances the minimization of the total loss and the maximization of the margin. An SVM is formulated as in Problem (4):

$$\min_{w,b} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{l=1}^{n} \max\left\{1 - y_i\left(\mathbf{w}^T \boldsymbol{x}_i\right), 0\right\} \tag{4}$$

where $\mathbf{w}$ denotes the hyperplane parameters and C is the constant for the maximization of the margin, which can be tuned via training.

Problem (4) can be reformulated as Problem (5) to use the gradient descent algorithm to train the FCN and SVMs synchronously:

$$\begin{aligned} &\min_{w,b} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{l=1}^{n} \xi_i \\ &\text{s.t.} \quad y_i\left(\mathbf{w}^T \boldsymbol{x}_i - b\right) \geq 1 - \xi_i \qquad i = 1, \ldots, n \\ &\qquad \xi_i \geq 0 \qquad\qquad\qquad\qquad\quad i = 1, \ldots, n \end{aligned} \tag{5}$$

2.2 | Uncertainty framework

In this section, an uncertainty framework is incorporated into the FCN to handle the uncertainties in features and labels. The methods against uncertainty in features and labels are presented in Section 2.2.1 and Section 2.2.2, respectively. The fusion method against the two uncertainties is described in Section 2.2.3.

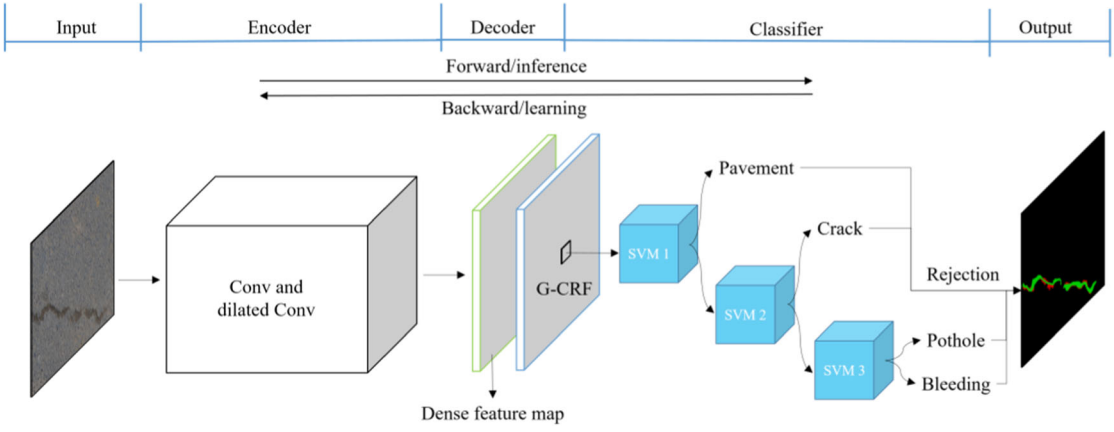


FIGURE 4 Architecture of the proposed FCN

2.2.1 | Uncertainty in features

Uncertainty in the features mainly arises due to the indistinguishability between defects and noise patterns in the data set (Zhang et al., 2017) and the unreasonable structure of the FCN, among other factors. It derives from the epistemic errors. The objective of the uncertainty framework for the features is to provide a method for training a feature-robust FCN model that considers such epistemic uncertainties and realizes high accuracy.

The uncertainty in the features is modeled in each training data element $\boldsymbol{x}_i$ as $\Delta\boldsymbol{x}_i$, where $\Delta\boldsymbol{x}_i \in \mathbb{R}^p$. The features in the training data set take values $(\boldsymbol{x}_i+\Delta\boldsymbol{x}_i)$ for $i = 1, \ldots, n$. Thus, the uncertainty in the features is defined as the following set:

$$u_x = \left\{ \Delta\mathbf{X} \in \mathbb{R}^{n\times p} \left| \left\|\Delta\boldsymbol{x}_i\right\|_q \le \rho, i = 1,2,\ldots,n \right. \right\} \quad (6)$$

where ρ is the degree parameter of the considered perturbation of the features in the training data set, which is used to represent the deviation of the features from their nominal values. ρ can be determined via accuracy optimization of the FCN, which is discussed in Section 4.1.2. The FCN without the feature uncertainty is equivalent to the nominal FCN if $\rho = 0$. Hence, the FCN with the feature uncertainty is preferred over the nominal FCN if ρ is reasonable. Additionally, u_x is the Cartesian product, as shown in Equation (6). Thus, it can also be used to consider alternative uncertainty sets for feature uncertainties in future works. In this study, the norm uncertainty set u_x is considered as an example and we apply it in practice in Section 4.

As discussed in Section 2.1.3, the energy $E(\mathbf{x})$ is imported into the SVMs. Therefore, Problem (5) should be reformulated as Problem (7) to take the uncertainty set into account.

$$\min_{\mathbf{w},b} \max_{\Delta\mathbf{X}\in u_x} \sum_{i=1}^{n} \max\left\{1 - y_i\left(\mathbf{w}^T\left(\boldsymbol{x}_i + \Delta\boldsymbol{x}_i\right) - b\right), 0\right\} \quad (7)$$

Referring to the study of Xu, Caramanis, and Mannor (2009), the robustness counterpart to Problem (7) is reformulated as Problem (8).

$$\begin{aligned} &\min_{\mathbf{w},b} \sum_{i=1}^{n} \xi_i \\ &\text{s.t.} \quad y_i\left(\mathbf{w}^T\boldsymbol{x}_i - b\right) - \rho\|\mathbf{w}\|_{q*} \qquad i = 1,\ldots,n \end{aligned} \quad (8)$$

where $\|\mathbf{w}\|_{q*}$ is the dual norm of $\mathbf{w}$. In Problem (8), ξ_i is determined by $\mathbf{w}$, b, y_i, and ρ. The sum of ξ_i can be minimized via the training of the SVMs and the FCN, for which the procedure is the same as for the FCN model.

2.2.2 | Uncertainty in labels

The uncertainty in labels is one of the most important problems in pixel-level classification. It occurs naturally in the labels that are specified by humans because pixel-level labeling is time-consuming, complex, and semiobjective. It is an aleatory uncertainty. A scenario is considered in which a subset of the pixel-level labels deviates from the true classes.

Since aleatory uncertainty occurs randomly, it is modeled in a part of the training data set in $\boldsymbol{y}_i$ as Δy_i, where $\Delta\boldsymbol{y}_i \in \{0,1\}$; $\Delta\boldsymbol{y}_i$ is equal to 1 if the label deviates from the true class and 0 otherwise. Thus, the uncertainty in the labels is defined as the following set:

$$u_y = \left\{ \Delta\mathbf{y} \in \{0,1\}^n \left| \sum_{i=1}^{n} \Delta y_i \le \lambda, i = 1,2,\ldots,n \right. \right\} \quad (9)$$

where λ is a positive integer that corresponds to the considered random deviation in the labels, which represents the deviation of the labels from their nominal values. λ can be determined via the accuracy optimization of the FCN, which is discussed in Section 4.1.2. The FCN with uncertainty in the labels is equivalent to the nominal FCN if $\lambda = 0$. Hence, the FCN model with uncertainty in the labels is preferred over the nominal FCN if λ is reasonable. Unfortunately, u_y is not a Cartesian product. Thus, the true labels in the training data set are modeled as $y_i(1 - 2\Delta y_i)$.

As discussed in Section 2.1.3, the labels for the FCN are compared with the outputs of the three SVMs to compute the loss function. Therefore, Problem (5) is reformulated as Problem (10) to take the aleatory uncertainty set into account.

$$\min_{\mathbf{w},b} \max_{\Delta\mathbf{y}\in u_y} \sum_{i=1}^{n} \max\left\{1 - y_i\left(1 - 2\Delta y_i\right)\left(\mathbf{w}^T\boldsymbol{x}_i - b\right), 0\right\} \quad (10)$$

The robustness counterpart to Problem (10) is Problem (11):

$$\begin{aligned} \min\ & \sum_{i=1}^{n} \xi_i + \lambda q + \sum_{i=1}^{n} r_i \\ \text{s.t.}\ & q + r_i \ge \phi_i - \xi_i && i = 1,\ldots,n \\ & \xi_i = \max\left\{1 - y_i\left(\mathbf{w}^T\boldsymbol{x}_i - b\right), 0\right\} && i = 1,\ldots,n \\ & \phi_i = \max\left\{1 + y_i\left(\mathbf{w}^T\boldsymbol{x}_i - b\right), 0\right\} && i = 1,\ldots,n \\ & r_i, \phi_i, \xi_i \ge 0 && i = 1,\ldots,n \\ & q \ge 0 \end{aligned} \quad (11)$$

In Problem (11), the minimization of r_i depends on $\mathbf{w}$, b, y_i, and λ. The sum of r_i over i is minimized via the training of the SVMs and the FCN, for which the procedure is the same as for the FCN model.

2.2.3 | Uncertainty in both features and labels

In this section, the methods in Section 2.2.1 and Section 2.2.2 are integrated into the FCN model to provide a new classifier

that is stable under uncertainty in both features and labels. The integration can be considered as an extension of the two previous uncertainty frameworks to handle the errors in both features and labels in the training data set.

Naturally, Problem (5) is reformulated as Problem (12) to consider u_x and u_y, where ρ and λ are the degree parameters of the considered perturbation of the features and the positive integer of the considered random deviation of the labels in the training data set, respectively. If $\rho = 0$ or $\lambda = 0$, the FCN is equivalent to the FCN without the uncertainty framework. Then, the robustness counterpart to Problem (12) is Problem (13):

$$\min_{\mathbf{w},b} \max_{\Delta \mathbf{y} \in u_y} \max_{\Delta \mathbf{X} \in u_x} \sum_{i=1}^{n} \max\left\{1 - y_i\left(1 - 2\Delta y_i\right) \times \left(\mathbf{w}^T\left(\boldsymbol{x}_i + \Delta \boldsymbol{x}_i\right) - b\right), 0\right\} \tag{12}$$

$$\begin{aligned} \min \ & \sum_{i=1}^{n} \xi_i + \lambda q + \sum_{i=1}^{n} r_i \\ \text{s.t.} \ & q + r_i \geq \phi_i - \xi_i && i = 1, \ldots, n \\ & \xi_i = \max\left\{1 - y_i\left(\mathbf{w}^T \boldsymbol{x}_i - b\right) + \rho \|\mathbf{w}\|_{q*}, 0\right\} && i = 1, \ldots, n \\ & \phi_i = \max\left\{1 + y_i\left(\mathbf{w}^T \boldsymbol{x}_i - b\right) + \rho \|\mathbf{w}\|_{q*}, 0\right\} && i = 1, \ldots, n \\ & r_i, \phi_i, \xi_i \geq 0 && i = 1, \ldots, n \\ & q \geq 0 \end{aligned} \tag{13}$$

In Problem (13), the minimization of r_i depends on $\mathbf{w}$, b, y_i, ρ, and λ. The sum of r_i over i is minimized via the training of the SVMs and the FCN, whose procedure is the same as for the FCN model.

2.3 | Probability-based rejection

Falsely identified areas are always identified in the segmentation results (Long et al., 2015), such as highlighted background areas in object areas. The decision rule of the classifier is one of the causes. In an FCN, a probability distribution for each pixel is provided by the SVMs. The class with the maximal probability is regarded as the class of the pixel. Sometimes, this leads to a mistake if the maximal probability is not predominant. Therefore, a rejection rule based on the Bayes rule is proposed to avoid this scenario in this study.

The confidence metric, which is denoted as $CM(H_j)$, is defined in Equation (14):

$$CM\left(H_j\right) = \frac{P\left(p_1 p_2 \cdots p_M | H_j\right) P\left(H_j\right)}{\sum_{i=1}^{M} P\left(p_1 p_2 \cdots p_M | H_i\right) P\left(H_i\right) + P\left(p_1 p_2 \cdots p_M | H_{M+1}\right) P\left(H_{M+1}\right)} \tag{14}$$

where H_j is the assumption that a pixel belongs to class j, with $j \in \{Background, Crack, Pothole, Bleeding, Conflict\}$. "*Conflict*" denotes that the output probability distribution of the proposed FCN cannot indicate to which class the pixel belongs and the distribution is equivocal. M is equal to 4 in this study. $P(H_i)$ and $P(H_{M+1})$ are the prior probabilities of class j and "*Conflict,*" respectively. In this study, we assume that the prior probabilities are equal.

By considering the independence of each class, the posterior probability is simplified and computed via Equations (15) and (16):

$$P\left(p_1 p_2 \cdots p_M | H_j\right) = \prod_{i=1}^{M} P\left(p_i | H_j\right) \ j = 1, \ldots, M+1 \tag{15}$$

$$P\left(p_k | H_j\right) = \frac{1}{n} \sum\nolimits_n Ker\left(p_k - a_k^j\right) \ j = 1, \ldots, M \tag{16a}$$

$$Ker\left(x\right) = \frac{1}{\sqrt{2\pi}h} \exp\left(-\frac{x^2}{2h^2}\right) \tag{16b}$$

$$P\left(p_k | H_{M+1}\right) = \frac{1}{M-1} \sum_{k \neq j} P\left(p_k | H_j\right) \tag{16c}$$

where h is the standard deviation in the Gaussian distribution; p_k is the probability of a pixel belonging to class k, which is specified by the SVMs.

Once a rejection threshold TH has been specified, a rejection decision can be made via Equation (17):

$$Rejection \Rightarrow \max_{j=1,\ldots,M} CM\left(H_j\right) + \frac{CM\left(H_{M+1}\right)}{M} \leq 1 - TH \tag{17}$$

2.4 | Model with FCN-GCRF, the uncertainty framework, and probability-based rejection

The FCN-GCRF with the uncertainty framework and probability-based rejection is combined as illustrated in Figure 5. The loss function of the model in Figure 5 is defined as

$$Loss = \frac{1}{H}\left(\sum_{i=1}^{H} \sum_{j=1}^{C} 1\left\{y^i = j\right\} \log \frac{e^{a_{i,j}^{l+2}}}{\sum_{l=1}^{C} e^{a_{i,l}^{l+2}}}\right) \tag{18}$$

where $1\{y^i = j\}$ is the logical expression. The loss function will return 1 if a predicted class of the ith sample is true, 0 if a predicted class is false, and 0.25 ($1/M$) if a rejection decision has been made.

In addition, pixel-level accuracy is also used to measure the performance of the algorithm during the training as

$$mPA = \frac{1}{n} \sum_{i=1}^{n} \frac{p_r}{p_t} \tag{19}$$

where n is the number used in the training or validation. p_r is the number of pixels in an image that are classified correctly. p_t is the total number in the image.

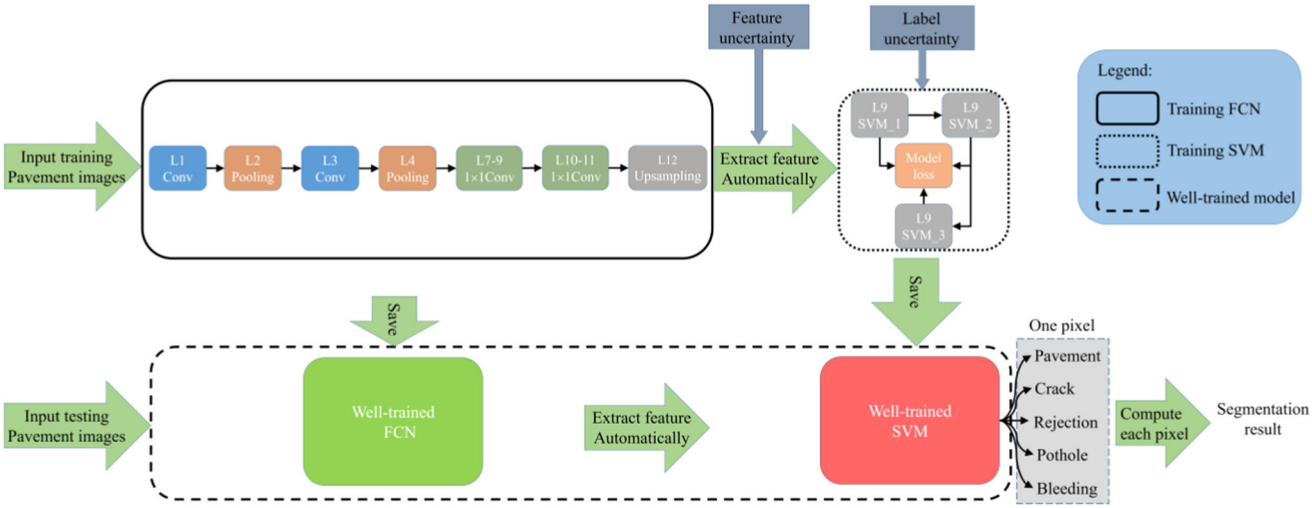


FIGURE 5 Implementation process of the end-to-end combination model

3 | DATA SET AND IMPLEMENTATION DETAILS

In this section, the database for the proposed combination is presented in Section 3.1. The implementation details for the combination model are presented in Section 3.2.

3.1 | Image collection

The asphalt pavement sections that are used to generate a database on which to train and test the proposed algorithm are located in four provinces in China, as shown in Figure 6. A Datong SH5047XJCA2D5 multifunction inspection vehicle was used to collect pavement images. The camera lens was kept perpendicular to pavements, and the distance between the camera and the pavements was invariant in acquiring images. Pavement digital image collection was conducted on 16 asphalt pavements with various service lifetimes, conditions, traffic volumes, and mixture types to guarantee the completeness of the database. The resolutions of collected pavement images are in the range of 72–300 pixels per inch. The ratio between the distress length in images and the actual length is 1:2 in this work.

3.2 | Implementation details

A total of 8,820 pavement defect images with various resolutions and sizes are acquired. In the proposed method, the sizes of the input images are not fixed. Thus, the original images are directly used to generate a database. Randomly, 5,292 and 1,764 pavement images were selected for the training and validation data sets, respectively. The remainder of the database was utilized as a testing data set. The corresponding ground-truth images of pavement defects are labeled at the pixel level manually. Pavement, cracks, potholes, and bleeding are labeled in a Python environment, and there is no rejected area in the ground-truth images. Figure 7 shows several examples.

As the uncertainty framework and probability-based rejection are not included in the Caffe (Jia et al., 2014), it is recompiled as discussed in Section 2. During each training iteration, images are fed into the network in mini-batches of size 16. To accelerate the convergence and avoid the overfitting, logarithmically decreasing learning rates are adopted with a base learning rate of 0.01. The momentum and weight decay are 0.9 and 0.005, respectively. Models with various values of ρ and λ are trained using the recompiled Caffe for 60,000 iterations and saved every 500 iterations to evaluate their accuracy on the validation data set. The above procedures are performed on a computer that is equipped with an Intel® Core™ i7-8750H CPU, 32.00 GB RAM, and an NVIDIA GeForce GTX 1080 8 GB GPU.

4 | RESULTS AND DISCUSSION

4.1 | Training performance

4.1.1 | Training performance of various models

Figure 8 shows the losses of the validation data set in models #1–#4 during the training processes, which are computed via Equation (18). The differences in models #1–#4 in Figure 8 are summarized in Table 1. ρ and λ in models #3 and #4 are 0.03 and 0.04. *TH* in model #4 is 0.25. Model #1 has been verified as a state-of-the-art technique for current crack detection systems (Dung & Anh, 2019; Yang et al., 2018). The previous studies (Cha et al., 2017; Tong, Gao, & Zhang, 2017) prove that the supervised methods (e.g., CNNs and FCNs) are more superior in pavement distress detection than the unsupervised methods, and only the supervised methods are considered in this study.

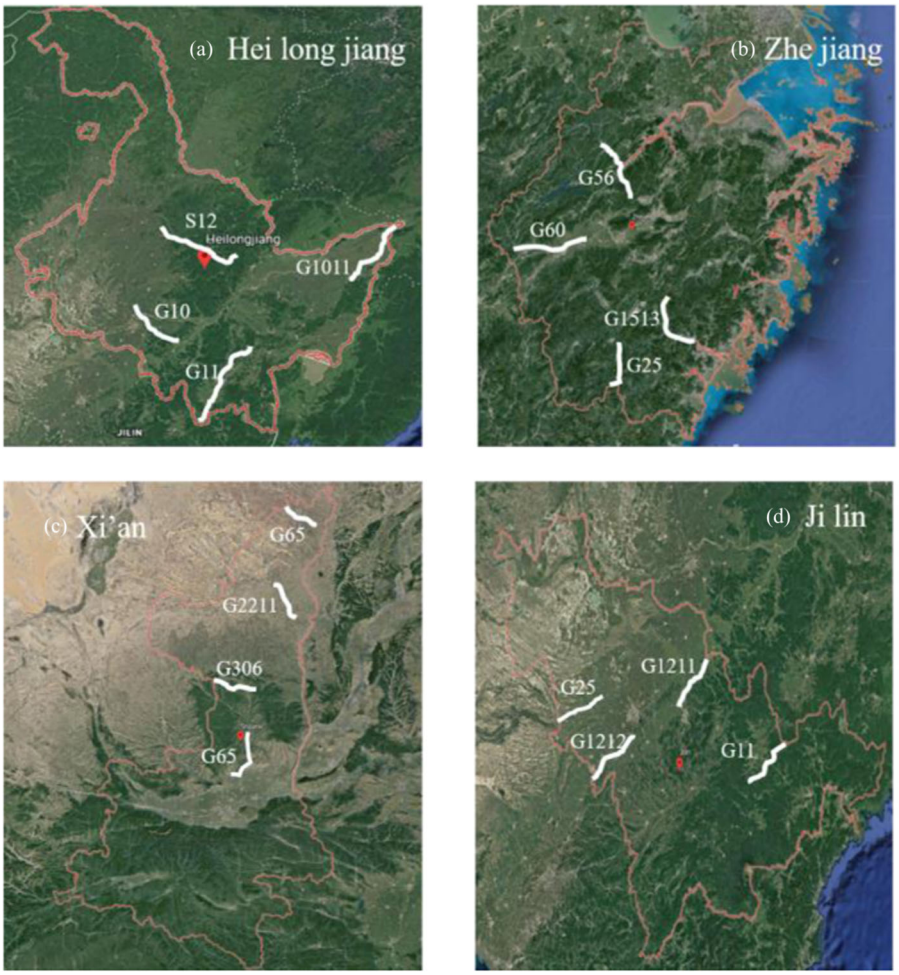


FIGURE 6 Asphalt pavement sections in China. The maps are supported by Google Earth

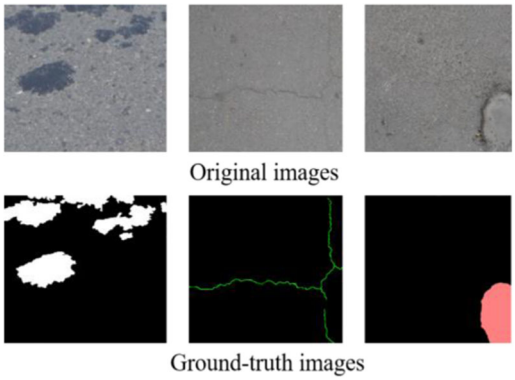


FIGURE 7 Examples in the database

Figure 8 shows that the final losses of models #1 and #2 were 0.140 and 0.096, respectively. Hence, the out-of-sample accuracy and generalization performance are improved by replacing the fully connected layers and a softmax layer with the SVMs. This is mainly because large weights and biases of the fully connected layers were required during the training, which results in overfitting. However, SVM classifiers try to find a hyperplane of imported features that maximizes the margin of separation using only a normal vector $\mathbf{w}$ and a displacement term b (Equation (6)). Thus, an SVM has fewer parameters than a fully connected layer, although the dimension of $\mathbf{w}$ is always higher than the dimension of the imported features. Reasonable simplified parameters are essential for preventing overfitting (Srivastava et al., 2014). The validation results of models #2 and #3 demonstrate that the uncertainty framework improves the proposed algorithm in terms of out-of-sample accuracy. Hence, there are errors in the labels and features of the training data set due to data corruption, which negatively affect the fitting during training. The label errors are due to pixel-level classification differences for defect areas among humans, especially for the defect boundaries. Eight people performed the labeling for over 3 months. The feature errors derive from the noise patterns in the database and the inherent drawbacks of the FCN-based methods. The utilization of the uncertainty framework enables this to be overcome. It provides an effective way of avoiding data corruption and feature derivation. Additionally, the fluctuation of losses in model #3 is less than those of models #1 and #2. This is

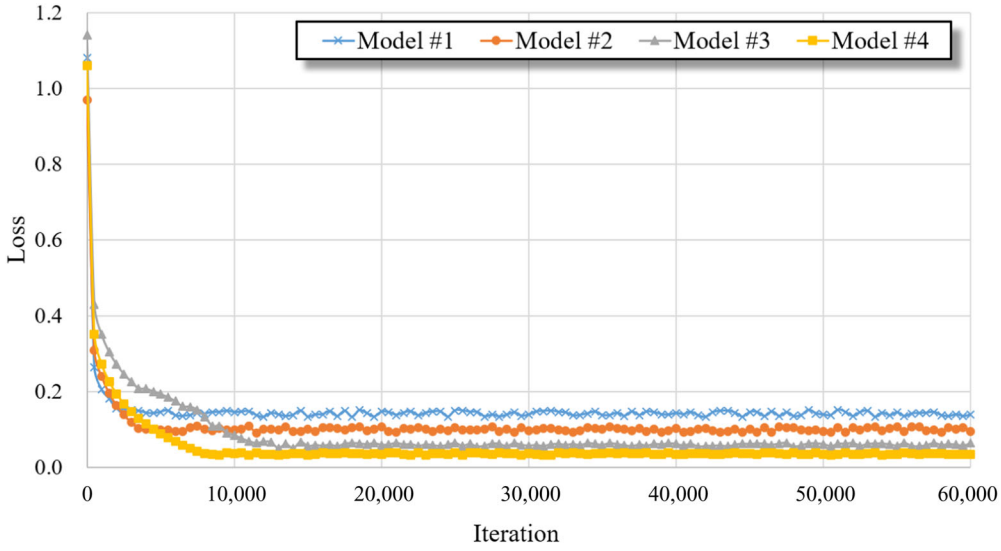


FIGURE 8 Segmentation loss errors of the deep-learning models

also mainly due to the uncertainty framework. In deep learning, the parameter λ can also be regarded as a regularization term for controlling the weight iteration. By comparing models #3 and #4, the losses are reduced further. With the use of probability-based rejection, the condition that the maximal probability is not predominant is removed. We can conclude that various pixel-level classifications with equivocal probability distributions are rejected in model #4. Therefore, compared with current crack detection systems based on deep learning (Dung & Anh, 2019; Yang et al., 2018), the proposed process (model #4) improves the accuracy and generalization capacity for pavement detection systems.

Figure 9 presents the pixel-level classification accuracies of the four models. The rejected pixels are regarded as half-correct classifications. The accuracies are bounded from above by the training accuracy and from below by the validation accuracy. Model #4 realized the highest average accuracy, while the average accuracy of model #3 is also acceptable. The gaps for models #3 and #4 between the training and validation curves are less narrow than the ones of the other two models. Hence, the accuracy and generalization performance of a deep-learning model can be improved by utilizing the uncertainty framework and probability-based rejection. In addition, models #3 and #4 realized small standard deviations among the classes. This is because the two models fully consider the uncertainty of the boundaries between pavements and defects, while models #1 and #2 do not. The boundaries between pavements and defects are fuzzy and vary among humans.

In addition, compared with the sum of the individual efficacies, the integration of an FCN with a G-CRF and SVMs, an uncertainty framework, and probability-based rejection have two advantages: (a) the parameters in the three parts can be adjusted simultaneously, which is promising to improve the accuracy, and (b) an end-to-end method can be conducted by the integration, which processes the raw data directly to avoid the information loss among the three parts.

4.1.2 | Parametric studies on training samples

Since the training data set used in this study is not as large as a typical training data set (Everingham, Van Gool, Williams, Winn, & Zisserman, 2010), a parametric study on training samples is conducted to determine whether the database is sufficient. Training data sets of the following sizes are generated: 1,058 (20%), 2,117 (40%), 2,646 (50%), 3,175 (60%), 3,704 (70%), 4,234 (80%), 4,763 (90%), and 5,292 (100%).

Figure 10 presents the validation results of the parametric studies. The required number of pavement images is at least 4,234 for models #3 and #4. However, the required numbers for models #1 and #2 are 5,292 and 4,763, respectively.

TABLE 1 Differences in models #1–#4

Number	G-CRF	SVM classifier	Uncertainty framework	Probability-based rejection
Models #1	√	×	×	×
Models #2	√	√	×	×
Models #3	√	√	√	×
Models #4	√	√	√	√

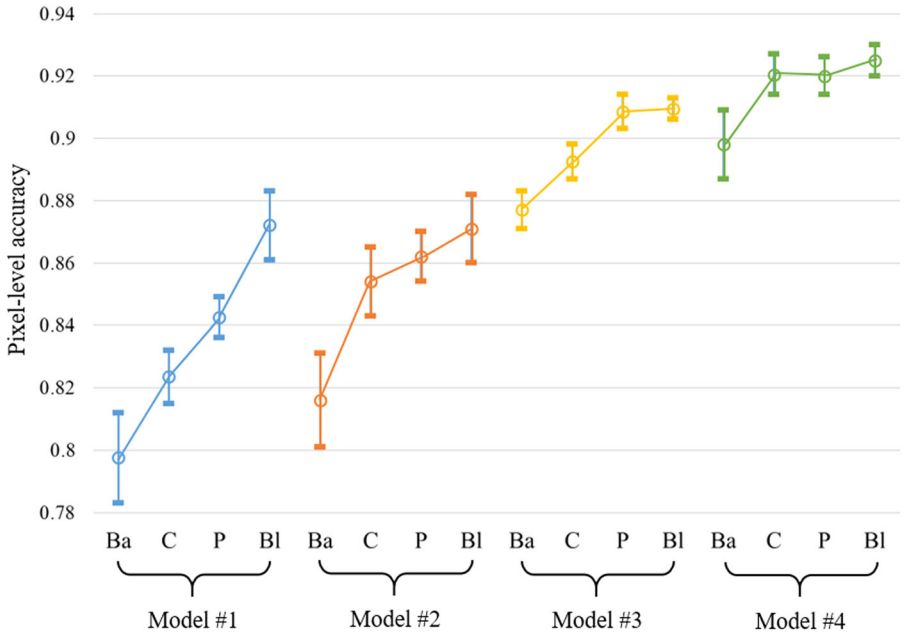


FIGURE 9 Pixel-level classification accuracies of the deep-learning models. Ba, C, P, and Bl denote background, crack, pothole, and bleeding, respectively

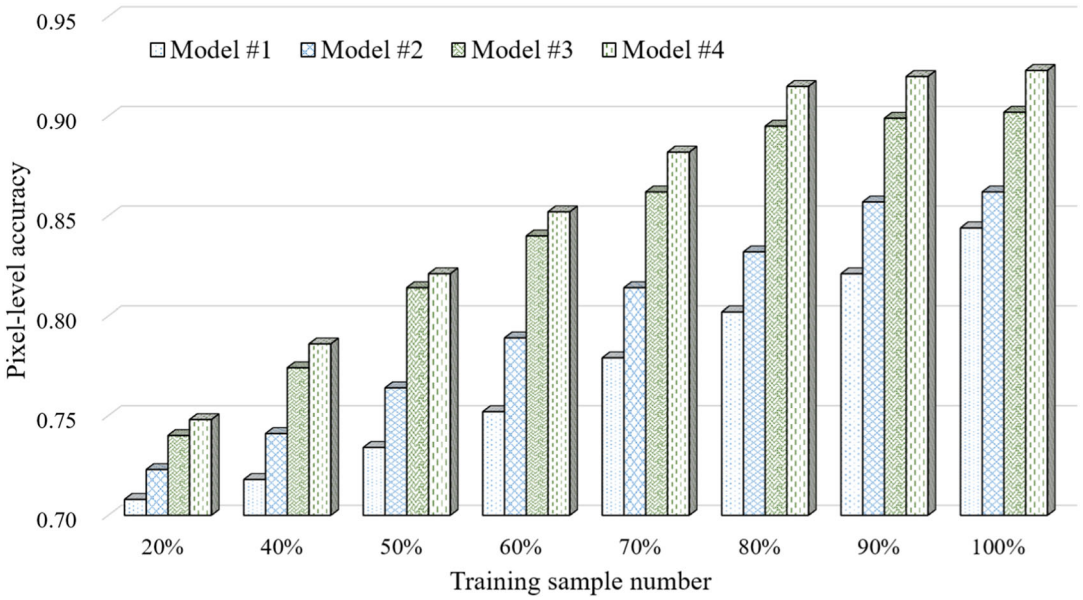


FIGURE 10 Validation results of parametric studies

Hence, the generalization performances and out-of-sample accuracies of the deep-learning methods are improved by the combination with the SVMs and the uncertainty framework. It provides a reasonable approach for developing a deep-learning pixel-level classifier with high out-of-sample accuracy and generalization performance using a training data set that is smaller than a typical training data set. In addition, the training data set in this study is sufficient for developing the proposed method.

4.1.3 | Training performances with various uncertainties

As discussed in Section 2.2, reasonable values of parameters ρ and λ can improve the accuracy of the models theoretically. In Section 4.1.1 and Section 4.1.2, we consider the models with ρ of 0.03, λ of 0.04, and TH of 0.25. The optimality of the training and validation performances cannot be verified, though the performances are acceptable. In practice, the feature and label uncertainties can be determined by varying the

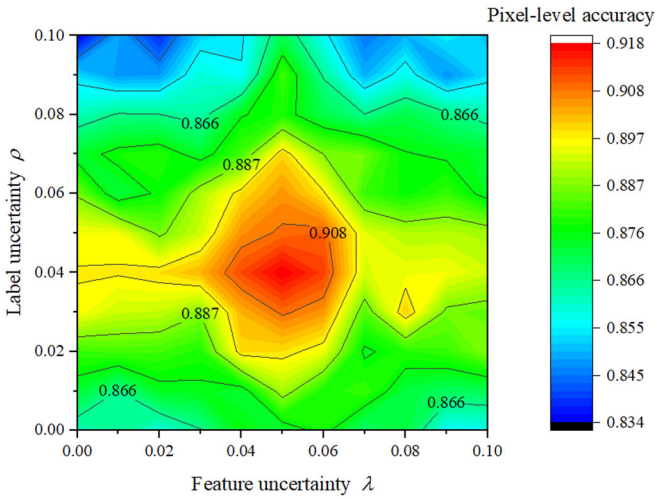


FIGURE 11 Validation performance of model #4 with various feature and label uncertainties

values of λ and ρ. *TH* can be determined from the rejection percentage and the error rejection percentages.

Figure 11 presents the pixel-level accuracy of model #4 with various feature uncertainties and label uncertainties. Only the validation results of the validation data set are reported in Figure 11. The probability-based rejection of model #4 is not considered in Figure 11. The optimal values of ρ and λ are 0.05 and 0.04. The validation accuracy of the model with ρ equal to 0.05 and λ equal to 0.04 is 91.81%. The validation accuracy of the model without ρ or λ is 86.17%. The uncertainty framework can improve the pixel-level accuracy. By varying the values of λ and ρ, the optimal uncertainty parameter values can be determined in the application for pavement defect detection.

In Figure 11, as λ increases, the accuracy of the model increases initially and subsequently decreases. Hence, label uncertainty exists in the training data set. The real label uncertainty is approximately 0.05; hence, approximately 0.04 of the pixel labels in Section 4.1.1 are not correct. As ρ increases, the accuracy of the model increases initially and subsequently decreases. Hence, feature uncertainty exists in model #4. The real feature uncertainty is approximately 0.03. Therefore, the structure of the convolution and pooling layers in the model is not optimal but reasonable. By comparing the effects of the two parameters, it concludes that the different judgments for defect areas among people have a larger effect than the structure of the convolution and pooling layers. This is because the pixel-level accuracy increases from 87.25% to 91.18% as the value of λ increases from 0 to 0.04, while the accuracy increases from 89.75% to 91.18% as the value of ρ increases from 0 to 0.05.

Figure 12 presents the rejection percentages, error rejection percentages, and correct rejection percentages of model #4 with different *TH*s at the pixel level. Error rejection refers to a rejection action for a pixel that was classified correctly. The model with ρ equal to 0.05 and λ equal to 0.04 is reported in Figure 12. With the decrease of *TH*, model #4 tends to reject more pixels; hence, increasingly many predicted probability distributions are regarded as blurry by the probability-based rejection criterion. As the number of rejections increases, model #4 makes fewer incorrect pixel-level segmentations. However, it can also be found that the error rejection percentages increase with the decrease of *TH*. Hence, model #4 rejects much correct pixel-level segmentation, which is not expected in the task. Thus, a reasonable *TH* value is necessary. To balance the error and correct rejection percentages, a *TH* value of 0.25 is optimal. The model with *TH* equal to 0.25 rejects sufficiently many incorrect pixel-level segmentations but makes a small number of error rejections.

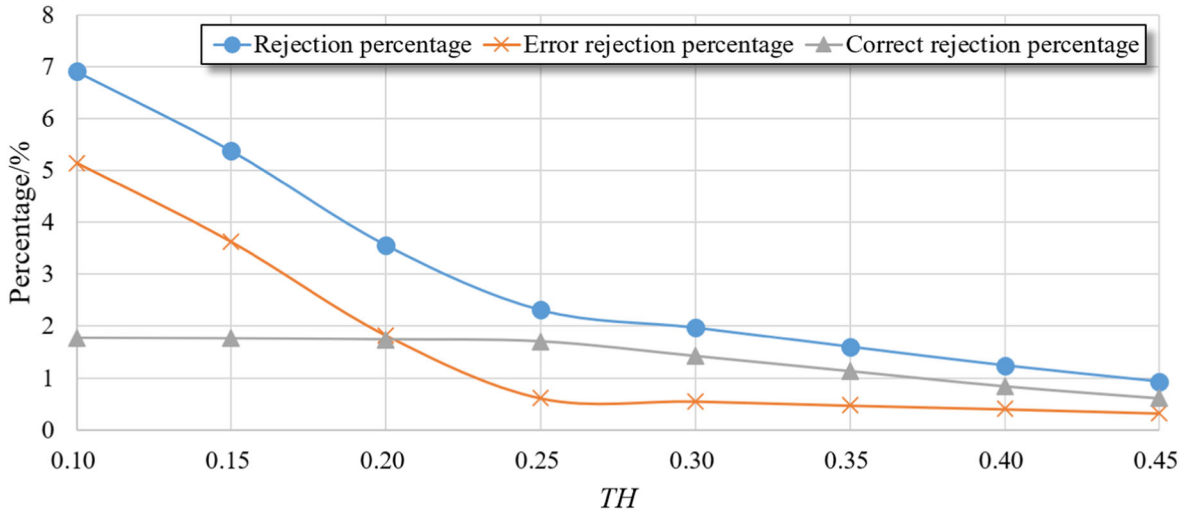


FIGURE 12 Rejection performance of model #4 with ρ equal to 0.05 and λ equal to 0.04

TABLE 2 Overall performances of four models in the evaluation

	mPA	mIoU	fwIoU
Model #1	0.782	0.722	0.882
Model #2	0.801	0.736	0.908
Model #3	0.813	0.745	0.914
Model #4	0.822	0.758	0.921

4.2 | Testing performance

4.2.1 | Overall performance

To further evaluate the performance of the optimal model, the testing data set and three indices are used (Long et al., 2015). The indices are the mean pixel accuracy (mPA), the mean intersection over union (mIoU), and the frequency-weighted IoU (fwIoU), which are expressed in Equations (20)–(22):

$$mPA = \frac{1}{k}\sum_{i=0}^{k}\frac{p_{ii}}{\sum_{j=0}^{k} p_{ij}} \tag{20}$$

$$mIoU = \frac{1}{k+1}\sum_{i=0}^{k}\frac{p_{ii}}{\sum_{j=0}^{k} p_{ij} + \sum_{j=0}^{k} p_{ji} - p_{ii}} \tag{21}$$

$$fwIoU = \frac{1}{\sum_{i=0}^{k}\sum_{j=0}^{k} p_{ji}}\sum_{i=0}^{k}\frac{p_{ii}}{\sum_{j=0}^{k} p_{ij} + \sum_{j=0}^{k} p_{ji} - p_{ii}} \tag{22}$$

where k is the number of pavement defects; p_{ii} is a true positive; p_{ij} is a pixel that belongs to class i but is predicted to belong to class j, namely, a false positive; and p_{ji} is a false negative.

Table 2 presents the overall performances of the four models in the evaluation. Parameters ρ and λ in models #3 and #4 are 0.04 and 0.05, respectively. TH in model #4 is 0.25. The rejected areas are not regarded as incorrect segmentations. According to Table 2, the out-of-sample accuracy and the generalization performance are improved by the SVMs, the uncertainty framework, and the probability-based rejection. The reasons why the SVMs and the uncertainty framework improve the out-of-sample accuracy and the

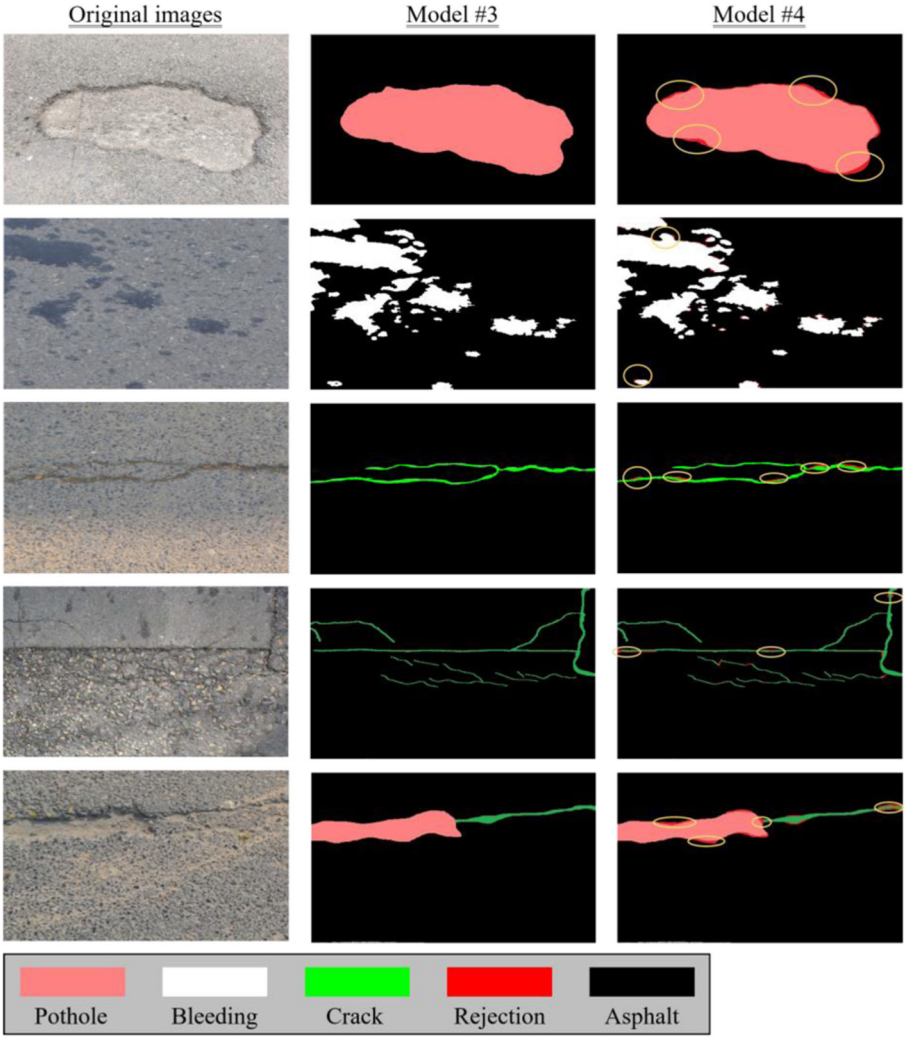


FIGURE 13 Examples of segmentation results. Various rejected areas are circled by yellow circles for clarity

generalization performance have been discussed in Section 4.1.1. Probability-based rejection improves the performance of model #4 by removing false segmentation, as shown in Figure 13. The results in Figure 13 indicate that model #4 has similar accuracy levels in asphalt pavements with different colors. For example, the results of the distress segmentation in rows 2, 3, and 5 are in similar accuracy levels, although these pavements have different colors. As the features related to pavement distresses in asphalt and concrete pavements are similar, model #4 has the potential to have similar accuracy levels to detect concrete pavement distresses after fine-tuning (Li, Grandvalet, & Davoine, 2018). Additionally, it prefers to reject areas around the defects, which are the boundaries between defects and asphalt. The definition of the boundaries is blurry and varies from one labeler to another. Hence, SVMs sometimes output plausible probability distributions and cause incorrect classifications and segmentations. This phenomenon is avoided by probability-based rejection. Additionally, pavement inspectors can analyze the development tendencies of pavement defects based on the distribution of the rejected areas.

Although the proposed combination realizes reasonable pavement defect detection performance, there remain incorrect segmentations in the testing data set. This is due

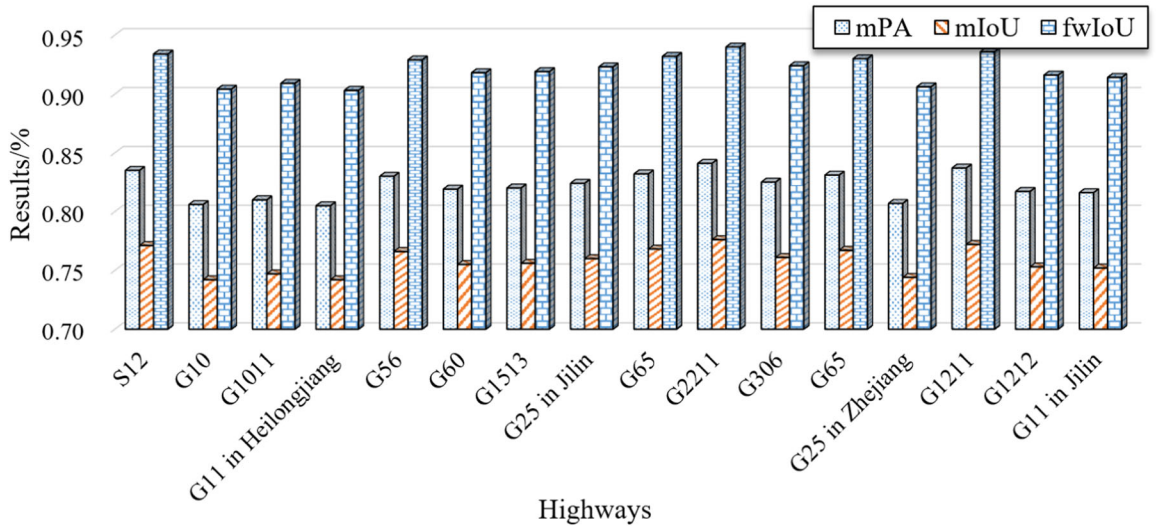


FIGURE 14 Performance of model #4 on various pavements

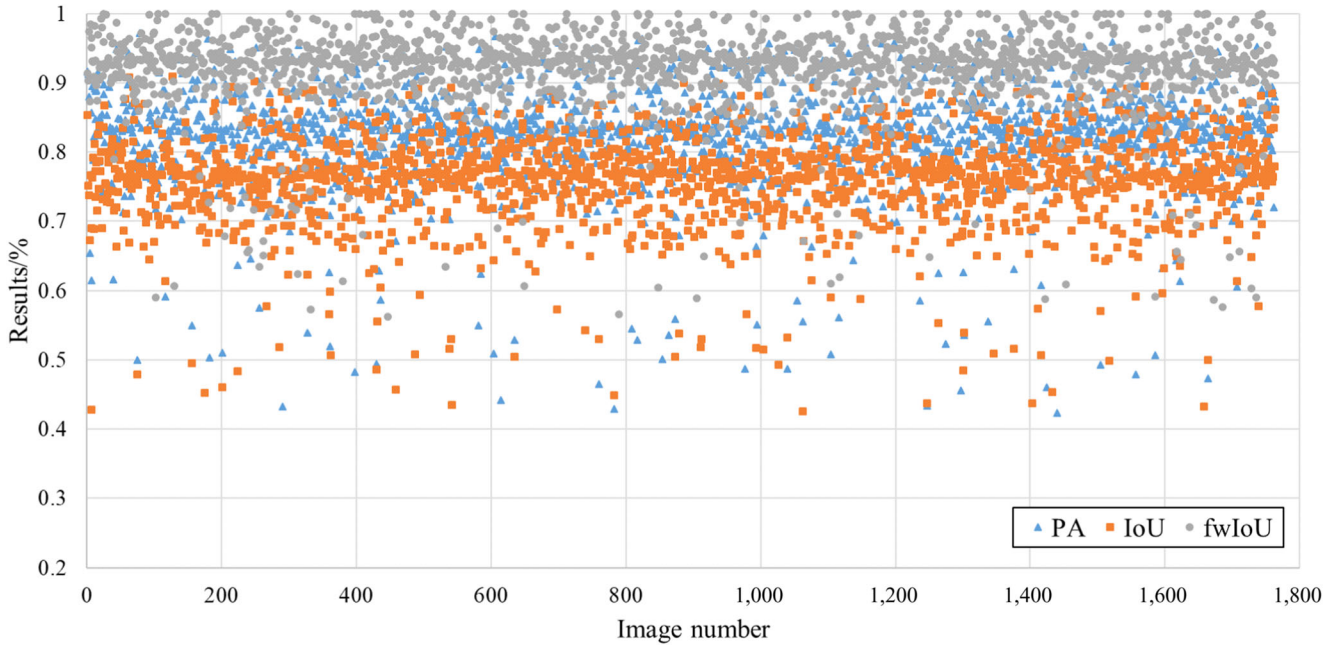


FIGURE 15 Plot of the PAs, IoUs, and fwIoUs of model #4

to the defect similarity, such as the shape similarity between potholes and bleeding areas. In addition, some crack features are not evident in the pavement images. These factors lead to feature uncertainty. For example, the pavement distresses in Figure 13 are well segmented except the crack on the far right at the top in row 3. The features of the pavement in this part are similar to the features of the crack, such as textures. It is even difficult for humans to distinguish the crack from the pavements. In the view of deep learning, it leads to the confusing feature extraction in the convolutional and pooling layers. The confusing features are used for the pixel-level classification in SVMs. With the use of the confusing features the algorithm is not able to capture the crack. The tolerance of the feature uncertainty framework is limited. In addition, some defects always occur in the same areas, which leads to a label uncertainty in the database. The tolerance of the label uncertainty framework is also limited. Therefore, a more reasonable label formulation should be adopted to overcome this problem.

4.2.2 | Performance on various pavements

In pavement engineering, it is important to verify that model #4 performs well on the pavements with various service lifetimes, conditions, traffic volumes, and mixture types. The testing data set is divided according to the pavements to study the pavement stability of model #4. The mPAs, mIoUs, and fwIoUs of the 16 pavements are plotted in Figure 14. The PAs, IoUs, and fwIoUs that are obtained by model #4 for the 1,764 testing images are plotted in Figure 15. Parameters ρ, λ, and TH in model #4 are 0.05, 0.04, and 0.25, respectively. Figures 14 and 15 show that model #4 has similar mPAs, mIoUs, and fwIoUs among pavements. Hence, the stability of model #4 is acceptable. The end-to-end deep-learning model can detect pavement defects in various pavements because the many kernels in the convolution layers can extract many low-, mid-, and high-level features related to pavement defects automatically.

4.2.3 | Processing time for a real-time system

The proposed combination is expected to be embedded in a multifunction testing vehicle system. Thus, the computation time should be as low as possible. The models #1–#4 are implemented using Caffe on a computer that is equipped with an Intel® Core™ i7-8750H CPU, 32.00 GB RAM, and an NVIDIA GeForce GTX 1080 8 GB GPU.

The computation times of the four models are listed in Table 3. The speed of model #4 is ideal, and no detection delay is introduced by the SVMs, the uncertainty framework, or the probability-based rejection algorithm. Therefore, model #4 can be used for a real-time detection system. The processing times of the four models are similar because the complexity of the whole calculation is not increased by the SVMs, the uncertainty framework, or the probability-based rejection. Hence, the proposed combination improves the precision of the supervised algorithm without substantially increasing the computational complexity or computation time. In addition, a model parallel computation (X. W. Chen & Lin, 2014) can be used to solve the problem of the continuous improvement of image resolutions in recent years. We conducted a parallel computation with two GPUs. The computation times decrease obviously, as shown in Table 3.

TABLE 3 Computation times of four models for defect detection (unit: s/image)

Mode	Model #1	Model #2	Model #3	Model #4
GPU	0.124	0.152	0.153	0.162
GPU parallel	0.076	0.088	0.092	0.103

5 | CONCLUSIONS AND RECOMMENDATIONS

In this study, a novel combination for defect detection using pavement digital images is presented. The performance of the proposed combination is discussed in detail. The following conclusions can be drawn from the presented results and discussion:

1. The proposed combination, integrated with GCRF, the uncertainty framework, and probability-based rejection, realizes an mPA of 0.822, an mIoU of 0.758, and an fwIoU of 0.921 in the evaluation for pavement defect detection. The out-of-sample accuracy, the generalization performance, and the pavement stability of the combination are acceptable. The combination can be utilized to replace onsite inspections by humans partially and to provide the defect information in its entirety.
2. The generalization performance and out-of-sample accuracy of an FCN-based model can be improved by replacing fully connected layers and the softmax layer by SVMs to avoid overfitting because an SVM has fewer parameters than a fully connected layer or a softmax layer.
3. The problems of epistemic and aleatory uncertainties exist in the pavement defect detection. The integration of the uncertainty framework using ρ and λ can reduce the negative effects of the two uncertainties and improve the accuracy and generalization performance of the proposed combination for pavement defect detection.
4. The probability-based rejection approach improves the performance of model #4 by removing unreasonable segmentations. Additionally, it prefers to reject areas in the boundaries between defects and asphalt. The blurry areas always lead SVMs to output plausible probability

distributions, which may cause incorrect segmentations. The phenomenon is avoided by probability-based rejection.

5. Compared with the sum of the individual efficacies, the integration of an FCN with a G-CRF and SVMs, an uncertainty framework, and probability-based rejection have two advantages: a simultaneous parameter adjusting in the three parts and an end-to-end method for processing the raw data to avoid information loss.
6. The computation time of the proposed combination is ideal, and the combination can be embedded into a multifunction testing vehicle system for real-time detection. The SVMs, uncertainty framework, and probability-based rejection algorithm do not introduce a detection delay because the complexity of the whole calculation is not increased by the integration.
7. The proposed integration provided a way to segment pavement defects from pavement digital images. The segmentation results can be used for defect measurement based on some state-of-the-art technologies in the future, for example, crack width measurement (W. Wang, Zhang, Wang, Braham, & Qiu, 2018).

ACKNOWLEDGMENTS

This work is supported by National Natural Science Foundation of China (No. 51978067), Science and Technology Development Project of Xinjiang Production and Construction Corps (No. 2019AB013), Key Research and Development Program of Shaanxi Province of China (No. 2019GY-174), and the Co-operation Program with the UTs and INSAs (France) funded by the China Scholarship Council (No. CSC201801810108).

REFERENCES

Abdel-Qader, I., Abudayyeh, O., & Kelly, M. E. (2003). Analysis of edge-detection techniques for crack identification in bridges. *Journal of Computing in Civil Engineering*, *17*(4), 255–263.

Adeli, H., & Samant, A. (2000). An adaptive conjugate gradient neural network–wavelet model for traffic incident detection. *Computer-Aided Civil and Infrastructure Engineering*, *15*(4), 251–260.

Ben-Tal, A., Bhadra, S., Bhattacharyya, C., & Nemirovski, A. (2012). Efficient methods for robust classification under uncertainty in kernel matrices. *Journal of Machine Learning Research*, *13*(10), 2923–2954.

Bertsimas, D., Brown, D. B., & Caramanis, C. (2011). Theory and applications of robust optimization. *Society for Industrial and Applied Mathematics Review*, *53*(3), 464–501.

Bertsimas, D., Dunn, J., Pawlowski, C., & Zhuo, Y. D. (2018). Robust classification. *INFORMS Journal on Optimization*, *1*, 1–90.

Bhattacharyya, C., Pannagadatta, K., & Smola, A. J. (2005). A second order cone programming formulation for classifying missing data. In *Proceedings of Advances in Neural Information Processing Systems* (pp. 153–160).

Bi, J., & Zhang, T. (2005). Support vector classification with input data uncertainty. *Advances in Neural Information Processing Systems*, *17*(1), 161–168.

Biggio, B., Nelson, B., & Laskov, P. (2011). Support vector machines under adversarial label noise. In *Proceedings of Asian Conference on Machine Learning* (pp. 97–112).

Cha, Y. J., Choi, W., & Büyüköztürk, O. (2017). Deep learning-based crack damage detection using convolutional neural networks. *Computer-Aided Civil and Infrastructure Engineering*, *32*(5), 361–378.

Cha, Y. J., Choi, W., Suh, G., Mahmoudkhani, S., & Büyüköztürk, O. (2018). Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types. *Computer-Aided Civil and Infrastructure Engineering*, *33*(9), 731–747.

Chandra, S., & Kokkinos, I. (2016). Fast, exact and multi-scale inference for semantic image segmentation with deep Gaussian CRFs. In *European Conference on Computer Vision* (pp. 402–418). Cham: Springer.

Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2017). DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *40*(4), 834–848.

Chen, S., Leng, Y., & Labi, S. (2019). A deep learning algorithm for simulating autonomous driving considering prior knowledge and temporal information. *Computer-Aided Civil and Infrastructure Engineering*. https://doi.org/10.1111/mice.12495

Chen, X. W., & Lin, X. (2014). Big data deep learning: Challenges and perspectives. *IEEE Access*, *2*, 514–525.

Daniel, A., & Preeja, V. (2014). A novel technique for automatic road distress detection and analysis. *International Journal of Computer Applications*, *101*(10), 18–23.

Dung, C. V., & Anh, L. D. (2019). Autonomous concrete crack detection using deep fully convolutional neural network. *Automation in Construction*, *99*, 52–58.

El-Sawy, A., Hazem, E. B., & Loey, M. (2016). CNN for handwritten arabic digits recognition based on LeNet-5. In *International Conference on Advanced Intelligent Systems and Informatics* (pp. 566–575). Cham: Springer.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The Pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, *88*(2), 303–338.

Farfade, S. S., Saberian, M. J., & Li, L. J. (2015). Multi-view face detection using deep convolutional neural networks. In *Proceedings of the 5th ACM International Conference on Multimedia Retrieval* (pp. 643–650). ACM.

Gao, Y., & Mosalam, K. M. (2018). Deep transfer learning for image-based structural damage recognition. *Computer-Aided Civil and Infrastructure Engineering*, *33*(9), 748–768.

Girshick, R. (2015). Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 1440–1448).

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2016). Region-based convolutional networks for accurate object detection and segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *38*(1), 142–158.

Guo, Y., Yao, A., & Chen, Y. (2016). Dynamic network surgery for efficient DNNs. *Proceedings of Advances in Neural Information Processing Systems* (pp. 1379–1387). CoRR, August.

Jarray, F., Boughorbel, S., Mansour, M., & Tlig, G. (2018). A step loss function based SVM classifier for binary classification. *Procedia Computer Science*, *141*, 9–15.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., … Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM International Conference on Multimedia* (pp. 675–678). ACM.

Kim, H., Ahn, E., Shin, M., & Sim, S. H. (2019). Crack and noncrack classification from concrete surface images using machine learning. *Structural Health Monitoring*, *18*, 725–738.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems* (pp. 1097–1105).

Laukkanen, T. (2016). Consumer adoption versus rejection decisions in seemingly similar service innovations: The case of the Internet and mobile banking. *Journal of Business Research*, *69*(7), 2432–2439.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

Li, X., Grandvalet, Y., & Davoine, F. (2018). Explicit inductive bias for transfer learning with convolutional networks. *arXiv preprint arXiv:1802.01483*.

Lin, J., & Liu, Y. (2010). Potholes detection based on SVM in the pavement distress image. In *2010 Ninth International Symposium on Distributed Computing and Applications to Business, Engineering and Science* (pp. 544–547). IEEE.

Liu, H., Sha, A., Tong, Z., & Gao, J. (2018). Autonomous microscopic bunch inspection using region-based deep learning for evaluating graphite powder dispersion. *Construction and Building Materials*, *173*, 525–539.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3431–3440).

McGrayne, S. B. (2011). *The theory that would not die: How Bayes' rule cracked the enigma code, hunted down Russian submarines, & emerged triumphant from two centuries of controversy*. London: Yale University Press.

Molina-Cabello, M. A., Luque-Baena, R. M., López-Rubio, E., & Thurnhofer-Hemsi, K. (2018). Vehicle type detection by ensembles of convolutional neural networks operating on super resolved images. *Integrated Computer-Aided Engineering*, *25*, 321–333.

Moskát, C., & Hauber, M. E. (2007). Conflict between egg recognition and egg rejection decisions in common cuckoo (*Cuculus canorus*) hosts. *Animal Cognition*, *10*(4), 377–386.

Natarajan, N., Dhillon, I. S., Ravikumar, P. K., & Tewari, A. (2013). Learning with noisy labels. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, & K. Q. Weinberger (Eds.), *Proceedings of Advances in Neural Information Processing Systems 26* (pp. 1196–1204). La Jolla, CA: Neural Information Processing Systems Foundation.

Nejad, F. M., & Zakeri, H. (2011). An optimum feature extraction method based on wavelet–radon transform and dynamic neural network for pavement distress classification. *Expert Systems with Applications*, *38*(8), 9442–9460.

Pant, R., Trafalis, T. B., & Barker, K. (2011). Support vector machine classification of uncertain and imbalanced data using robust optimization. In N. Mastorakis, V. Mladenov, Z. Bojkovic, F. Topalis, & K. Psarris (Eds.), *Proceedings of 15th WSEAS Internaternational Conference on Computers* (pp. 369–374). Stephens Point, WI: World Scientific and Engineering Academy and Society.

Rafiei, M. H., & Adeli, H. (2017). A novel machine learning-based algorithm to detect damage in high-rise building structures. *The Structural Design of Tall and Special Buildings*, *26*(18), e1400.

Rafiei, M. H., & Adeli, H. (2018). A novel unsupervised deep learning model for global and local health condition assessment of structures. *Engineering Structures*, *156*, 598–607.

Rafiei, M. H., Khushefati, W. H., Demirboga, R., & Adeli, H. (2017). Supervised deep restricted Boltzmann machine for estimation of concrete. *ACI Materials Journal*, *114*(2), 237–244.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems* (pp. 91–99).

Shen, F., Gan, R., Yan, S., & Zeng, G. (2017). Semantic segmentation via structured patch prediction, context CRF and guidance CRF. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognitio*n (pp. 1953–1961).

Shi, Z., Ye, Y., & Wu, Y. (2016). Rank-based pooling for deep convolutional neural networks. *Neural Networks*, *83*, 21–31.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15*(1), 1929–1958.

Sui, L., Feissel, P., & Denoeux, T. (2018). Identification of elastic properties in the belief function framework. *International Journal of Approximate Reasoning*, *101*, 69–87.

Sun, M., Song, Z., Jiang, X., Pan, J., & Pang, Y. (2017). Learning pooling for convolutional neural network. *Neurocomputing*, *224*, 96–104.

Tong, Z., Gao, J., Han, Z., & Wang, Z. (2018). Recognition of asphalt pavement crack length using deep convolutional neural networks. *Road Materials and Pavement Design*, *19*(6), 1334–1349.

Tong, Z., Gao, J., Sha, A., Hu, L., & Li, S. (2018). Convolutional neural network for asphalt pavement surface texture analysis. *Computer-Aided Civil and Infrastructure Engineering*, *33*(12), 1056–1072.

Tong, Z., Gao, J., & Zhang, H. (2017). Recognition, location, measurement, and 3D reconstruction of concealed cracks using convolutional neural networks. *Construction and Building Materials*, *146*, 775–787.

Tong, Z., Gao, J., & Zhang, H. (2018). Innovative method for recognizing subgrade defects based on a convolutional neural network. *Construction and Building Materials*, *169*, 69–82.

Tran, T. A., Jauberthie, C., Le Gall, F., & Travé-Massuyès, L. (2018). Evidential box particle filter using belief function theory. *International Journal of Approximate Reasoning*, *93*, 40–58.

Wang, K. C. P., & Smadi, O. (2011). *Automated imaging technologies for pavement distress surveys*. Transportation Research Circular No. E-C156, TRB, Washington DC: National Research Council.

Wang, L., Ouyang, W., Wang, X., & Lu, H. (2015). Visual tracking with fully convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 3119–3127).

Wang, M., & Cheng, J. C. (2019). A unified convolutional neural network integrated with conditional random field for pipe defect segmentation. *Computer-Aided Civil and Infrastructure Engineering*. https://doi.org/10.1111/mice.12481

Wang, N., Zhao, Q., Li, S., Zhao, X., & Zhao, P. (2018). Damage classification for masonry historic structures using convolutional neural networks based on still images. *Computer-Aided Civil and Infrastructure Engineering*, *33*(12), 1073–1089.

Wang, P., & Bai, X. (2018). Regional parallel structure based CNN for thermal infrared face identification. *Integrated Computer-Aided Engineering*, *25*(3), 247–260.

Wang, R., Li, W., Li, R., & Zhang, L. (2019). Automatic blur type classification via ensemble SVM. *Signal Processing: Image Communication*, *71*, 24–35.

Wang, W., Zhang, A., Wang, K. C., Braham, A. F., & Qiu, S. (2018). Pavement crack width measurement based on Laplace's equation for continuity and unambiguity. *Computer-Aided Civil and Infrastructure Engineering*, *33*(2), 110–123.

Xu, H., Caramanis, C., & Mannor, S. (2009). Robustness and regularization of support vector machines. *The Journal of Machine Learning Research*, *10*(7), 1485–1510.

Xue, Y., & Li, Y. (2018). A fast detection method via region-based fully convolutional neural networks for shield tunnel lining defects. *Computer-Aided Civil and Infrastructure Engineering*, *33*(8), 638–654.

Yamaguchi, T., & Hashimoto, S. (2010). Fast crack detection method for large-size concrete surface images using percolation-based image processing. *Machine Vision and Applications*, *21*(5), 797–809.

Yang, X., Li, H., Yu, Y., Luo, X., Huang, T., & Yang, X. (2018). Automatic pixel-level crack detection and measurement using fully convolutional network. *Computer-Aided Civil and Infrastructure Engineering*, *33*(12), 1090–1109.

Zhang, A., Wang, K. C., Li, B., Yang, E., Dai, X., Peng, Y., … Chen, C. (2017). Automated pixel-level pavement crack detection on 3D asphalt surfaces using a deep-learning network. *Computer-Aided Civil and Infrastructure Engineering*, *32*(10), 805–819.

Zhou, H., Zhang, J., Lei, J., Li, S., & Tu, D. (2016). Image semantic segmentation based on FCN-CRF model. In *2016 International Conference on Image, Vision and Computing* (pp. 9–14). IEEE.

Zou, Q., Cao, Y., Li, Q., Mao, Q., & Wang, S. (2012). CrackTree: Automatic crack detection from pavement images. *Pattern Recognition Letters*, *33*(3), 227–238.

How to cite this article: Tong Z, Yuan D, Gao J, Wang Z. Pavement defect detection with fully convolutional network and an uncertainty framework. *Comput Aided Civ Inf*. 2020;1–18. https://doi.org/10.1111/mice.12533