Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier DOI

Hierarchical Convolutional Neural Network with Feature Preservation and Autotuned Thresholding for Crack Detection

QIUCHEN ZHU[1], TRAN HIEP DINH[2,3], MANH DUONG PHUNG[1,3], and QUANG PHUC HA[1]

[1]School of Electrical and Data Engineering, University of Technology Sydney, Sydney, NSW 2000, Australia (e-mail: Qiuchen.Zhu@uts.edu.au; ManhDuong.Phung@uts.edu.au; Quang.Ha@uts.edu.au)

[2]UTS - VNU Joint Technology and Innovation Research Centre (JTIRC), Hanoi, Vietnam (e-mail: tranhiep.dinh@vnu.edu.vn)

[3]University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam

Corresponding author: Qiuchen Zhu (e-mail: Qiuchen.Zhu@uts.edu.au).

ABSTRACT Drone imagery is increasingly used in automated inspection for infrastructure surface defects, especially in hazardous or unreachable environments. In machine vision, the key to crack detection rests with robust and accurate algorithms for image processing. To this end, this paper proposes a deep learning approach using hierarchical convolutional neural networks with feature preservation (HCNNFP) and an intercontrast iterative thresholding algorithm for image binarization. First, a set of branch networks is proposed, wherein the output of previous convolutional blocks is half-sizedly concatenated to the current ones to reduce the obscuration in the down-sampling stage taking into account the overall information loss. Next, to extract the feature map generated from the enhanced HCNN, a binary contrast-based autotuned thresholding (CBAT) approach is developed at the post-processing step, where patterns of interest are clustered within the probability map of the identified features. The proposed technique is then applied to identify surface cracks on the surface of roads, bridges or pavements. An extensive comparison with existing techniques is conducted on various datasets and subject to a number of evaluation criteria including the average F-measure (AF_β) introduced here for dynamic quantification of the performance. Experiments on crack images, including those captured by unmanned aerial vehicles inspecting a monorail bridge. The proposed technique outperforms the existing methods on various tested datasets especially for GAPs dataset with an increase of about 1.4% in terms of AF_β while the mean percentage error drops by 2.2%. Such performance demonstrates the merits of the proposed HCNNFP architecture for surface defect inspection.

INDEX TERMS deep learning, crack detection, hierarchical convolutional neural network, feature preserving, thresholding.

I. INTRODUCTION

Surface inspection plays an important role in the health surveillance and hazard control of roads, bridges, pavements or tunnels. Effective maintenance and damage prevention of transport infrastructure rely on prompt detection for defects in transportation infrastructure such as cracks, edge failures, potholes, rutting, subsidence, or any surface deterioration [1]. For this, the inspection conducted manually by professional practitioners, wherein dangerous and unattainable sites would limit the effectiveness of human inspection. With advances in unmanned aerial vehicles (UAVs) and field robotics, machine vision-based systems are introduced to fulfill those inspection tasks [2]. For automatic inspection, successful identification of defect features on infrastructure surface requires the development of feasible, robust and effective detection algorithms.

In visual inspection from captured images, an intensity shift indicates a contrast between defective spots and their surrounding pixels in the color space. Based on the referred source of information, various methods for crack detection have been proposed. Initially, thresholding approaches are employed to execute fast detection by solely exploiting the statistics of intensity. Based on thresholding, early trials for surface imperfectness detection have been conducted with

hybrid utilization of intensity and geometrics [3]. Other methods roughly classify the image according to the distribution of pixel intensity [4]. To alleviate the interference from noisy textures, a scanning kernel like the Gabor filter has been developed in the frequency domain [5]. This scanning kernel can be used for pixel-wise extraction of the local geometric information about crack pixels and sharing the similarity to current convolutional kernels. To judge on surface defect, a probability model is usually formulated to determine the presence of cracks. An entropy formulation is introduced to guide the pavement crack detection based on saliency and statistical features [6]. Alternatively, Minimal Path Selection (MPS) on a single scale image [7] or multi-scale fusion [3] can be used in searching crack seeds. Promising results have suggested the formation of a trainable framework with improved robustness for crack detection.

With the increasing interest in artificial intelligence, machine learning (ML) approaches are introduced for heuristic abnormality detection. To this end, several ML techniques using shallow linear regression models like support vector machine (SVM) [8] and random forests [9] have been applied for crack detection. Such learning approaches provide an adaptive solution in disposing of a variety of crack patterns. However, the prediction accuracy may be limited by the model simplification and the available computational capacity in practice. On the other hand, deep convolutional neural networks (DCNN) [10] have been proposed as a probabilistic learning framework with a modest processing time that becomes very attractive for real-time applications. This technique provides an effective solution to challenges in semantic segmentation [11] owing to the capability of multiple-level abstraction and deep breakdown for identified features. Such promising results suggest the use of DCNN to identify a surface defect with binary segmentation.

In deep learning, often, the network is sequentially structured and finalized by fully-connected layers. Such structures may be computationally ineffective and cause blurred representation [12], leading to a drop in the prediction accuracy. Recently, the emerging hierarchical structure [13] has been applied to deal with the blurry problem. The potential of this framework in crack detection has been verified in [14]. In addition, a well-designed filter can be incorporated in the hierarchical convolutional learning process to extract the probability map of features [12]. Accordingly, it is promising to seek a suitable post-processing approach that can offer a more effective determination of crack and background with hierarchical DCNN.

Here, a hybrid approach is proposed, integrating a hierarchical convolutional neural network with feature preserving and the contrast-based autotuned thresholding (CBAT) technique to identify surface cracks of roads, bridges or pavements via aerial photography, obtained by a formation of unmanned aerial vehicles (UAV) [15]. The collected images are processed by the proposed neural network first for a probability map of a potential defect and then its features are further extracted by CBAT binarization for identification. Experiments on different datasets from [9] and on images captured by our UAVs during the inspection of a monorail bridge indicate the advantage of the hierarchical convolutional neural network with feature preservation (HCNNFP) proposed in comparison with some crack detection approaches available in the existing literature. Various frameworks for crack detection and specific datasets are considered in a number of experiments for comprehensive assessment on the merits of our HCNNFP as well as its improvement over other post-processing methods.

The paper is organized as follows. After the introduction, Section II discusses the relevant work for deep-learning-based crack detection. Section III describes the architecture of the proposed framework for crack detection. Section IV presents our thresholding technique for post-processing. Section V introduces our UAV system for image capturing, the datasets, the setup of two experiments respectively for comparison with relevant deep learning techniques and for post-processing with binarization. Section VI demonstrates the experimental results along with their discussion. Finally, a conclusion is drawn in Section VII.

II. RELEVANT WORK

In this section, key technologies using convolutional neural networks (CNN) in crack detection tasks are briefly discussed. Judging on a pipeline structure, CNN methods can be divided into sequential and hierarchical models. In sequential models, only the final output is involved in benchmark matching. For hierarchical models, features from multiple processing branches and the ground-truth can be utilized to collectively contribute to improving the fitness of the detection result.

Current CNN models for crack detection are listed in Table I, showing also the various methods that have been used to enhance the extracted features of the crack pattern. The sequential models include basic CNN [16], CNN with metaheuristics (CNN-M) [17], deep fully CNN (FCN) [18], Cracknet [19], Cracknet-V [20], and densely-connected CNN [21]. Those hierarchical CNN models relevant to this work are DeepCrack [14], feature pyramid and hierarchical boosting network (FPHBN) [22], U-Net [23], CNN with naïve Bayesian data fusion (NB-CNN) [24], weakly-supervised DCNN (WS-ConvNet) [25], PGA-Net [26] and SDD-Net [27]. Methods used for feature enhancement include deconvolutional decoders (D), residual modules (RM), probabilistic representation (PR), and statistic post-processing (SP). In the encoder-decoder structure, the crack patterns are to be rescaled with key indices recorded. With the preservation of those coordinates, the detailed patterns of crack features can be reasonably refilled with deconvolutional decoding. In RM, a combination of the original and processed features can be used to provide a residual effect like with human eyes, i.e., remembering the silhouette of the object that has previously been observed. The following step is to compensate for missing patterns using this residual effect. Alternatively, the feature maps are converted into a representation in the proba-

bilistic space PR. In this case, every pixel will be assigned a possibility score in the range of $(0, 1)$ to evaluate the likelihood of a crack. As a result, the prediction of the model becomes less overconfident on uncertainty and mislabels, contributing positively to the reduction of false-positive rate. The last approach SP using a global optimizer with statistic post-processing tools to extract the result from CNN. The accuracy of the detection can be improved by filtering out outliner labels with a total threshold or some verification mechanism.

Models	Type of architecture	D	RM	PR	SP
Basic CNN [16]	Sequential				
CNN-M [17]	Sequential				✓
FCN [18]	Sequential	✓			
CrackNet [19]	Sequential				
CrackNet-V [20]	Sequential				✓
Densely connected CNN [21]	Sequential		✓		✓
DeepCrack [14]	Hierarchical	✓	✓	✓	
FPHBN [22]	Hierarchical	✓	✓	✓	
U-Net [23]	Hierarchical	✓			
NB - CNN [24]	Hierarchical			✓	
WS-ConvNet [25]	Hierarchical		✓		✓
PGA-Net [26]	Hierarchical	✓	✓		
SDD-Net [27]	Sequential		✓		

TABLE I: Summary of CNN models with different architecture and different feature enhancement methods.

In this paper, the above enhancement methods are integrated to create a new processing pipeline for crack detection. The contributions of this paper can be summarized as follows:

- Among hierarchical architectures, the HCNNFP network proposed in this paper is different from the Deepcrack by a feature preserving branch. As such, it is more comprehensive in surface crack detection by using the combination of geometrical and statistic information, whereby feature abstraction is enhanced by an additional side branch in the encoder to reduce estimation error caused by redundant nonlinearity.
- An iterative approach is proposed to automatically search for an optimized threshold of the probability map for features generated from the proposed DCNN, and as a result, to avoid the time consumption in the search while increasing the accuracy of generated feature maps.
- A dynamic measure to evaluate the fitness of defect detection, assessing the average performance under a range of weights in conjunction with the commonly-used F-measure using a single pre-determined weight.

III. HIERARCHICAL CONVOLUTIONAL NEURAL NETWORK WITH FEATURE PRESERVATION

In this section, a novel DCNN approach called the hierarchical convolutional neural network with feature preservation (HCNNFP) is proposed to obtain a probability map of surface defects from the input image. Here, unlike the original hierarchical CNN, a feature preserving branch is augmented to adjust the weights of the abstraction from upper-layers, and hence, resolving the nonlinearity trade-off to improve the network performance.

A. CONVOLUTIONAL NEURAL NETWORKS

Our detection method is based on the inference in a convolutional neural network (CNN). To formulate the classification problem, let us first define a training sample as $D = \{(X, Y)\}$, where $X = \{x_{ij} | i, j \in (I \times J)\}$ and $Y = \{y_{ij} | i, j \in (I \times J)\}$ respectively represent the pixel values of the original image of size $I \times J$ and its corresponding annotated mask of cracks, both containing $I \times J$ pixels. In the context of defect identification, the ground-truth mask y_{ij} can take a binary value determined as,

$$y_{ij} = \begin{cases} 1, & x_{ij} \text{ - abnormal pixel in the mask,} \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

In a network model, the judgment on crack candidates is made from a layer-by-layer inference. Such inference is deduced from the basic structure of multilayer perceptrons (MLP) [28]. Suppose we have a L-layer MLP to predict the possibility map of defect candidates. For crack detection, the learning process targets at the best fitness to the annotated crack map. For an input matrix X^l in l-th layer, the learning process is an optimization problem formulated as

$$\begin{gathered} \min_{\{W^l\},\{b^l\}} \left\| P(X^L) - Y \right\|_F + \lambda \sum_l \left\| W^l \right\|_F^2 \\ \text{subject to } X^l = a\left(X^{l-1} W^l + b^l\right), l = 1, \ldots, L-1 \\ X^L = X^{L-1} W^L + b^L, \end{gathered} \tag{2}$$

where W^l and b^l represent the weights and bias at the l-th layer, $a(\cdot)$ is the nonlinear activation function, λ is the coefficient for controlling the scale of regularization, and $P(\cdot)$ is an arbitrary statistic function to express the possibility of crack candidates.

While CNN shares a similar architecture with that of MLP, it introduces the convolution to biologically stimulate the visual perception of cortex cells within a receptive field. By using the convolution operation $(*)$, the feature map of CNN can be expressed as a set of features:

$$f_{ij}^l = a\left(\left(W^l * X\right)_{ij} + b^l\right). \tag{3}$$

Then the target possibility $P(f_{ij})$ stimulates the conditional distribution $p(y_{ij}|x_{ij})$, the real possibility to indicate the confidence level of a pixel looking like a crack. However, it would be impractical for the neurons to be fully connected pixel-wisely due to the exponential increase of computational load. In this application, the non-zero weights are actually restrained within a certain kernel to present the logistic relation between a pixel and its limited neighborhood. The smallest kernel size that can provide the abstraction of the image is 3×3 (a 1×1 kernel can just output the original information). For a larger receptive field, the size of the kernel can be extended appropriately. Within the receptive field, the values of neighboring pixels collectively determine the intensity of a particular pixel on the feature map. As a result, the output can be a highly dependent abstraction of the crack patterns.

B. NETWORK STRUCTURE

The proposed symmetric network architecture is shown in Fig. 1, based on the framework of DeepCrack [14] with the same structural parameters such as the kernel size and the number of channels. The encoder consists of 5 convolutional blocks with 13 convolutional layers. Each convolutional layer contains a standard size-invariant convolution operation for the abstraction of crack features. With the combination of several convolutional layers, each feature map of the current scale is created at the end of each block. For downsampling, a max-pooling layer after each block compresses the image into quarter-size, preserving the values and indices of the local maxima. The shrinkage of the image consequently yields an increase in the size of the receptive field (RF) for the following layer, leading to a sparser feature map in the next block.

In the first two blocks, the convolutional kernels are relatively small compared to the size of the input. Accordingly, the first two blocks mainly preserve the detailed features of the original image. However, unlike the early CNNs aiming at small-size images as 28×28 in the Modified NIST dataset [29], the current deep learning approaches are designed to segment images with a size of at least hundreds by hundreds. Hence, the basic kernel in 3×3 is insufficient for abstraction, and as such, two convolutional layers are combined to stimulate the receptive field of a 5×5 kernel but with fewer parameters involved. The outputs from the third block are equal or less than the 1/16 size of the original image. Therefore, the receptive field of those blocks should be extended for preserving features. Consequently, an additional convolutional layer is added in each of those blocks. Here, a feature preserving branch is augmented at the output of the convolutional blocks in the encoder to adjust the level of abstraction from upper-layers by concatenation with the downsampling layer. Here, unlike the pipeline of the DeepCrack [14], which inherits the auto-encoder structure of SegNet [11], the proposed preserving branch is to maintain the image features by alleviating nonlinear redundancy, as explained at the end of this Section.

In our network, the decoder mirrors the structure of the encoder, with five corresponding upsampling layers to symmetrically retrieve the size of the image via the reference of saved indices. The sparse image generated from the last upsampling is refilled and reconstructed in the next blocks. With the index propagation throughout the entire pipeline, the network can restore key information of boundaries on the original image.

C. INFORMATION LOSS

Since the task of identifying a surface crack on a bridge, road or pavement can be rendered to a binary segmentation problem with two semantic groups, abnormal and intact features, the labeling error in the prediction can be evaluated by a binary entropy loss function. For the computation of a measure for information loss, let us define $F^k = \{f_{ij}^k | k = 1, \ldots, 5\}$ as the feature map under the zooming scale k and $F^{fused} = \{f_{ij}^{fused}\}$ as the fused map accordingly. The red modules depicted in Fig. 1 illustrates the formation of those feature maps. For an arbitrary feature f_{ij}, its pixel-wise probability $P(f_{ij})$ can be expressed by a sigmoidal function as,

$$P(f_{ij}) = \frac{1}{1 + e^{-f_{ij}}}. \tag{4}$$

In terms of binary entropy, a pixel-wise feature at a convolutional block is either abnormal or intact. By considering it as a random variable, the associated information loss for feature f_{ij}^k at the k^{th} convolutional block can be expressed via its entropy as,

$$l(f_{ij}^k) = -y_{ij} ln(P(f_{ij}^k)) - (1 - y_{ij}) \ln(1 - P(f_{ij}^k)). \tag{5}$$

Since the ground-truth mask contains only logical values 0 and 1, the information loss or entropy of Eq. (5) is rewritten as,

$$l(f_{ij}^k) = \begin{cases} -\ln P(f_{ij}^k), & y_{ij} = 1 \\ -\ln(1 - P(f_{ij}^k)), & y_{ij} = 0. \end{cases} \tag{6}$$

The accuracy of the prediction relies on the fitness of every feature map in comparison with the ground-truth mask. Hence, all the corresponding probability maps are responsible for the loss function, including for all fused pixels and all convolutional blocks. Accordingly, the total loss $\mathcal{L}$ of an image should represent the superposition of the pixel-wise losses for each convolutional block for all feature maps and the fused map for all pixels, i.e.

$$\mathcal{L} = \sum_{i=1}^{I} \sum_{j=1}^{J} \left(l(f_{ij}^{fused}) + \sum_{k=1}^{5} l(f_{ij}^k) \right). \tag{7}$$

D. PERFORMANCE ENHANCEMENT

For U-shape networks like the U-net [23], enhancement of features can be achieved with a comprehensive design to avoid ambiguously stacking additional channels. Here, a feasible structure is implemented with an alternative allocation inside the network. To analyze the performance improvement of the proposed HCNNFP, we consider the nonlinear nature of the network and then adjust its structure to simultaneously reduce nonlinearity while preserving the image features by resolving the trade-off between them.

1) Feature preservation versus nonlinearity

From the probabilistic view, the attribution of a pixel can be properly described by either an abnormal or intact pixel member corresponding to two random events EV_1 and EV_0, respectively. Specifically, EV_1 is the event that the sampled pixel belongs to the abnormal group and EV_0 is when it belongs to the intact area. Accordingly, given an observation on the pixel x_{ij}, two conditional probabilities are defined, namely the probability $P(EV_1|x_{ij})$ or $P(EV_0|x_{ij})$ that x_{ij} belongs to surface abnormality such as a crack or not. To identify a potential defect, let us consider the probability $P(EV_1|x_{ij})$. From Bayes's rule, the conditional probability

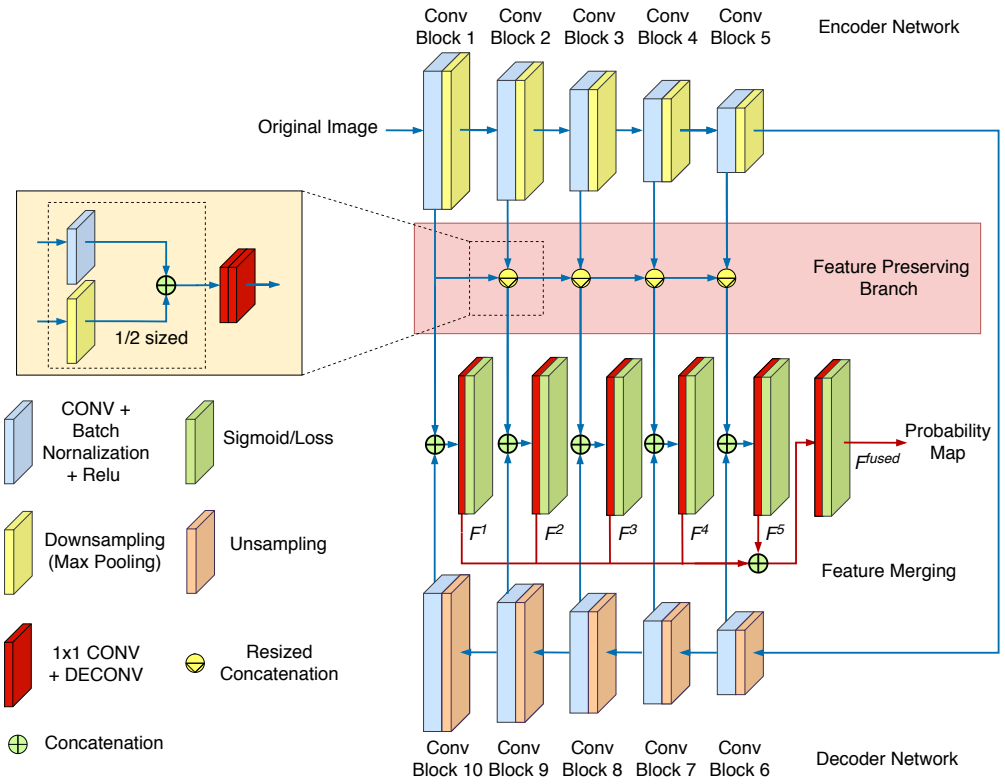

FIGURE 1: HCNNFP- network architecture

of a crack on a road, pavement or bridge, given pixel x_{ij} is expressed as

$$P\left(EV_1|x_{ij}\right) = \frac{P\left(EV_1, x_{ij}\right)}{P(x_{ij})},$$

or

$$\begin{aligned} P\left(EV_1|x_{ij}\right) &= \frac{P\left(x_{ij}|EV_1\right)P\left(EV_1\right)}{P\left(x_{ij}|EV_1\right)P\left(EV_1\right) + P\left(x_{ij}|EV_0\right)P\left(EV_0\right)} \\ &= \frac{1}{1 + \frac{P(x_{ij}|EV_0)P(EV_0)}{P(x_{ij}|EV_1)P(EV_1)}} \\ &= \frac{1}{1 + e^{-a(x_{ij})}}, \end{aligned} \tag{8}$$

where

$$a(x_{ij}) = \ln \frac{P\left(x_{ij}|EV_1\right)P\left(EV_1\right)}{P\left(x_{ij}|EV_0\right)P\left(EV_0\right)}. \tag{9}$$

Now, it is assumed that those conditional probabilities follow the Gaussian process $\mathcal{N}(\mu_{0,1}, \sigma^2)$ with the same variance σ [30] and means μ_1 and μ_0, respectively for the two abnormal and intact cases, we have:

$$P(x_{ij}|EV_{0,1}) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(x_{ij} - \mu_{0,1})^2}{2\sigma^2}\right], \tag{10}$$

in association with the random events EV_0 and EV_1. By substituting Eq. (10) into Eq. (9), the exponent $a(x_i)$ can be explicitly derived in the following form:

$$\begin{aligned} a(x_{ij}) &= \ln P\left(x_{ij}|EV_1\right) - \ln P\left(x_{ij}|EV_0\right) + \ln \frac{P\left(EV_1\right)}{P\left(EV_0\right)} \\ &= \frac{\mu_1 - \mu_0}{\sigma^2} x_{ij} + \frac{\mu_0^2 - \mu_1^2}{2\sigma^2} + \ln \frac{P\left(EV_1\right)}{P\left(EV_0\right)} \\ &= w x_{ij} + w_0, \end{aligned} \tag{11}$$

where $w = \frac{\mu_1 - \mu_0}{\sigma^2}$ and $w_0 = \frac{\mu_0^2 - \mu_1^2}{2\sigma^2} + \ln \frac{P(EV_1)}{P(EV_0)}$.

Therefore, by comparing Eq. (4) and Eq. (11), elements f_{ij} of an abnormal feature as of a crack can be expressed as,

$$f_{ij} = w x_{ij} + w_0, \tag{12}$$

which is linearly-dependent on the pixel values x_{ij} of the image captured on the surface under monitoring.

In a deep learning CNN framework, it is known that the hidden layers contain inevitable nonlinearity to facilitate the information processing capacity of the network. However, when using the sigmoidal function for probabilistic representation, the linear dependence of image features on the input appears to limit the amount of feature information throughout the processing. This requires a compromise between feature preserving and information handling. Due to the nonlinear activation employed in each convolutional layer, the overall nonlinearity accumulates largely in the forward propagation.

Therefore, the outputs from the deeper encoder network tend to depart further from the linear hypothesis, unfavorably influencing the accuracy of the probability maps as per Eqs. (8-12). Hence, some measures of compensation for nonlinearity is required to balance the trade-off between nonlinearity and the network capacity of information processing. This motivated us to develop a feature preserving branch(FPB) for the network architecture, for which a rationale is given in the following.

2) Feature preserving branch

By considering the benefits in the alleviation of redundant nonlinearity, here a side branch is created in the original HCNN to adjust the abstraction weights from the upper-layer by concatenation with the downsampling layer. As can be seen from the proposed network architecture of Fig. 1, a part of the encoder output passes by convolution, batch normalization and ReLU through an extra path and is half-sizedly concatenated with max-pooling in the downsampling. Furthermore, the concatenation takes place recursively between the output from the shallower encoder block and the output at the next deeper block. The input from each encoder block keeps semi-inherited in the feature merging routine to increase the possession of shallower-level features in merging channels along with the propagation in the convolutional network. A comparison of Deepcrack architecture and our proposed one is shown in Fig. 2, wherein learning performance can be significantly improved from resized concatenations in FPB so as not to increase the computational latency.

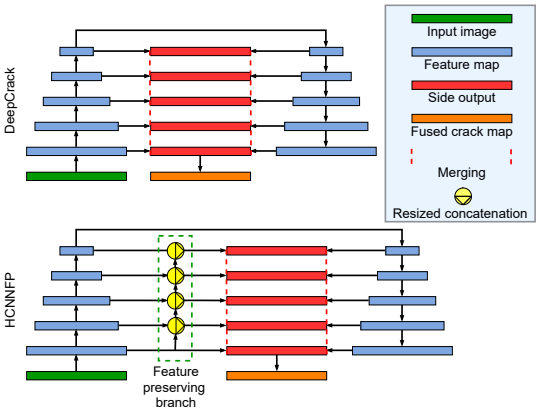

FIGURE 2: Architecture comparison

Notably, in our proposed enhanced HCNN network, the encoder outputs place higher weights on the feature maps from shallow layers, which result in (i) feature preservation via reduced nonlinearity as per Eq. (12), (ii) maintenance of the performance of information processing via deep learning, and (iii) assurance of the same level of computational complexity via half-sized concatenation. Under the premise of overall nonlinearity reduction and merits of the proposed approach, this adjustment can enhance the quality of the probability maps, and as such, increase the accuracy of crack detection for automatic monitoring of built infrastructure for transportation.

IV. CONTRAST-BASED AUTOTUNED THRESHOLDING

The output of the proposed pipeline is a probability map represented by a sigmoidal function. To obtain the binary map of images captured for surface inspection, it is required a suitable threshold for a fitting match to the image ground truth. The threshold, used to categorize the probability map into two classes, should be adaptable to various scenarios of surface imagery for accurate detection of cracks. To this end, an autotuning iterative thresholding technique is proposed to obtain the best threshold based on image contrast and subject to various thresholding criteria.

A. PROBABILITY MAP THRESHOLDING CRITERIA

In this paper, the following criteria are used for evaluation.

F-measure (F_β): The probability map is binarized by a threshold $T \in (0, 1)$, where pixels with a probability greater (smaller) than T are categorized as imperfect or intact regions. For a single image, the score F_β [31] is often used as a fitness criterion to justify the chosen threshold. For a well-selected threshold, such F-measure is expected to reach the maximum. It is calculated as

$$F_\beta = (1 + \beta^2) \cdot \frac{p_T \times r_T}{\beta^2 \times p_T + r_T}, \tag{13}$$

where p_T and r_T are respectively the precision and recall, based on the correctly-reported and falsely-reported positive or negative results; and β denotes the weight between precision p_T and recall r_T. A larger F-measure indicates a higher performance of the thresholding. In industrial practice, precision plays an important role in further disposal since p_T score represents the ratio between the identified defect over and the return features. Such information is quite valuable to the decision on the scope of maintenance or repair work to remedy the identified defect.

Average F-measure (AF_β): To emphasize the precision over recall r_T, the weight β^2 should be chosen less than 1. Especially, when $\beta^2 = 1$, F_β becomes the standard F-measure with equal weighting on the precision and recall [31], which can be expressed as $F_1 = 2 \times IoU/(1+IoU)$ and considered as mathematically alternative to the Intersection-over-Union (IoU) metric itself [32]. As recommended in [33], coefficient β^2 could be selected at 0.3. On other hand, β^2 = 0.25 is also frequently used to evaluate the quality of image processing [34]. However, no strong evidence is demonstrated in the literature to prove the priority of 0.25 or 0.3 among other adjacent values. For a fairer comparison through the F-measure, we propose a new evaluation metric to calculate the average F-measure AF_β over a given range of β^2. Here, the average F-measure, AF_β, is formulated by:

$$AF_\beta = \frac{1}{\beta_2^2 - \beta_1^2} \int_{\beta_1^2}^{\beta_2^2} F_\beta d\beta^2, \tag{14}$$

where β_1^2 and β_2^2 represent are respectively the lower and upper limit for the interested range of weight β^2. Substituting Eq. (13) to Eq. (14), AF_β can be explicitly obtained as

$$AF_\beta = r_T + \frac{r_T}{p_T} \times \frac{p_T - r_T}{\beta_2^2 - \beta_1^2} \times \ln \frac{p_T\beta_2^2 + r_T}{p_T\beta_1^2 + r_T}. \quad (15)$$

Notably, this metric is considered as robust over any range of interest for weight β^2. Here, β^2 is nonzero while the condition that $\beta^2 < 1$ should be kept as in common practice. In terms of evaluation judgment, similarly to F_β, a higher AF_β represents better quality of a thresholding technique.

MAE: For a binary map $S = \{s_{ij}\}$, the mean absolute error (MAE) can be obtained from the post-processing step as,

$$MAE = \frac{1}{I \times J} \sum_{i=1}^{I} \sum_{j=1}^{J} |s_{ij} - y_{ij}|. \quad (16)$$

A smaller *MAE* indicates a better match to the ground truth (GT). Complementarily to the weighted F-measure, this metric rewards predictions with a high recall rate due to its preference for false-positive samples [35]. Alternatively, one can also use the mean absolute percentage error (MAPE) [36] with the total number of pixels in the image being replaced by the number of true-positive pixels N_{tp} in the denominator of MAE as,

$$MAPE = \frac{1}{N_{tp}} \sum_{i=1}^{I} \sum_{j=1}^{J} |s_{ij} - y_{ij}| \quad (17)$$

to make the results more salient.

B. INTERCONTRAST ITERATIVE THRESHOLDING

An even threshold $T = 0.5$ is usually considered as a reasonable value for good thresholding. However, this may cause mislabeling in the case with an unbalanced ratio between a faulty feature and an intact background. To obtain a better result, it is worth seeking a mechanism for autotuning of the threshold. To this end, we propose the contrast-based autotuned thresholding (CBAT), a contrast-based approach refined from Otsu's thresholding [37], to improve the accuracy of binarization. A flowchart for obtaining the binary map is depicted in Fig. 3, wherein Otsu's thresholding is only implemented in the region of interest (ROI) that encompasses a cluster of high-probability defect candidates during the iteration process rather than a large background region.

In the initial iteration, the whole histogram is predefined as the original ROI R_{ROI}^0. Generally, in the m^{th} iteration, Otsu's algorithm $otsu(.)$ obtains a threshold T_{ROI}^m for the previous ROI R_{ROI}^{m-1} such that

$$otsu(R_{ROI}^{m-1}) = T_{ROI}^m. \quad (18)$$

Threshold T_{ROI}^m is expected to segment R_{ROI}^{m-1} into region of interest R_{ROI}^m and background R_b^m, i.e.

$$R_{ROI}^{m-1} = R_{ROI}^m \cup R_b^m, \quad (19)$$

where R_{ROI}^m is the current ROI containing the pixels with a probability higher than T_{ROI}^m, and R_b^m is the corresponding background region for pixels whose probability is between T_{ROI}^m and T_{ROI}^{m-1}.

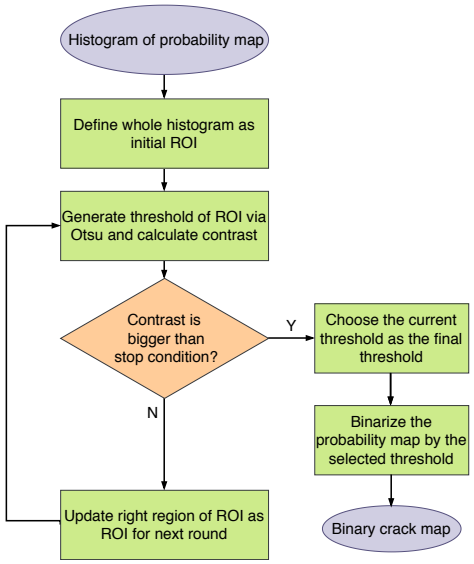

FIGURE 3: Binarization flowchart using CBAT

The interclass contrast [38] is a criterion for the assessment of segmentation quality, under the hypothesis that the intensity of homogenous pixels is close to the average intensity of their class. Referred to the probability histogram, the interclass contrast C_{ROI}^m for region R_{ROI}^{m-1} is expressed as,

$$C_{ROI}^m = \frac{|\,\mu_{ROI}^m - \mu_b^m\,|}{\mu_{ROI}^m + \mu_b^m}, \quad (20)$$

where μ_{ROI}^m and μ_b^m are respectively the mean probability in R_{ROI}^m and R_b^m. Due to a significant reduction of the ROI pixel number, the sum $\mu_{ROI}^m + \mu_b^m$ remains decreasing with iterations. Consequently, C_{ROI}^m keeps increasing until the iteration terminates. A strong contrast implies an obvious difference between the two classes within the probability map, resulting in a distinction between abnormal features and the intact region on the image captured.

Since our target is to highlight an imperfect region as a crack from its neighborhood, a proper interclass contrast is required to preserve the discernibility of defect candidates. For a particular surface type, the termination condition for contrast C_s is set so that the loop will stop when $C_{ROI}^m > C_s$ to yield the ultimate threshold T_u. The pseudo-code for the proposed CBAT approach is demonstrated in Algorithm 1.

The ROI, initially the whole histogram, is reduced after running the Otsu's algorithm for the first time at threshold T_1, then keeps shrinking iteratively with an updated interclass contrast (C_{ROI}^m to result in a new region of interest during the next searches for T_2, T_3, ... until reaching the termination condition for the ultimate value T_u. The probability region with a threshold T greater than T_u will be assigned to defect candidates, and the remaining pixels will be categorized as the background.

Algorithm 1 Intercontrast Iterative Thresholding

Require: R_i^0: whole histogram
Ensure: T_u: ultimate threshold
$m \leftarrow 0$
repeat
  $m++$
  $T_{ROI}^m \leftarrow otsu(R_{ROI}^{m-1})$
  $R_{ROI}^m \leftarrow R_i^{m-1}(R_{ROI}^{m-1} > T_{ROI}^m)$
  $R_b^m \leftarrow R_{ROI}^{m-1}(T_{ROI}^{m-1} \leq R_{ROI}^{m-1} < T_{ROI}^m)$
  $\mu_{ROI}^m \leftarrow Average(R_{ROI}^m), \mu_b^m \leftarrow Average(R_b^m)$
  $C_{ROI}^m \leftarrow \mid \mu_{ROI}^m - \mu_b^m \mid /(\mu_{ROI}^m + \mu_b^m)$
until $(C_{ROI}^m > C_s)$
$T_u \leftarrow T_{ROI}^m$

V. EXPERIMENTS AND EVALUATION

The UAV inspection system used for collecting images includes two subsystems: Skynet for flying drones and capturing images, and the ground center for data processing. The quadcopters are controlled to follow an IoT-communicated formation [39] to inspect a monorail bridge, as depicted in Fig. 4. Here, all the trainable parameters are initialized by He Normal initialization [40]. The training is conducted at a learning rate of $1e$-5 and optimized by Adaptive Moment Estimation (Adam) [41] with the default setting of two hyperparameters, (0.9, 0.999). For performance improvement from using the proposed loss function, the maximum training epoch is set as 30, which is adequately large for the sake of convergence. An early stop is applied when the reduction of the loss between two adjacent epochs is under 0.01%. The training process is conducted on NVIDIA Tesla T4 GPUs 16Gb.

1) Datasets

Four datasets used in our experiments include:

- *Crack500* [22]: containing 500 images of pavement cracks with granular backgrounds in a unified size of 2560×1440 with a few samples under uneven illumination. Due to the limitation in GPU memory and computation power, all the images are resized to 256×256.
- *CrackForest* [9]: containing 118 images of road cracks with labeled masks in a size of 600×800 with a part of samples with the interference of shadow and painted marks. We rotate the images with a range from 0 to 90 degrees, flip them vertically and horizontally, and randomly crop the flipped images with a size of 256×256. Ten thousands augmented images are split into the training and the validation set with a ratio of 9:1. The rest 1180 images are preserved for testing.
- *DCD* [42]: containing 521 images of infrastructure cracks with texture and misleading marks under various light condition.
- *GAPs* [43]: containing 509 images of pavement cracks with densely granular backgrounds under poor light conditions. DCD and GAPs are both integrated into an unified size of 448×448, following [44].

Here, the original annotation of public datasets is kept for a fair comparison with peer methods. For images with larger sizes, a sliding window can be used to process the image clip by clip [45]. Our original dataset is also included for testing:

- *SYDCrack* [12]: With the inspection system introduced above, an image dataset is collected for some surface cracks on a monorail bridge with regularly textured backgrounds under a good illumination. Those images are collected at 15 locations where crack patterns are located. The integrated dataset contains 170 images, cropped into 850 pitches with a size of 224×224.

FIGURE 4: Bridge inspection

2) Benchmarking

In the first experiment, a comparative analysis is conducted between our proposed one and the recent state-of-the-art crack detection methods using deep learning. The frameworks for comparison are listed in the following:

- *HED* [46]: Retaining the encoder part of SegNet, the holistically nested edge detection (HED) merges feature maps from five different levels in the encoder. The last feature map is used for computation of the loss function.
- *RCF* [47]: Another edge detection technique for richer convolutional features (RCF) delivers a merged output from each convolutional layer in the encoder block while HED just outputs the final layer.
- *SegNet* [11]: This framework represents a standard end-to-end model with an auto-encoder.
- *DeepCrack* [14]: An end-to-end hierarchical network for crack extraction using the typical architecture of SegNet with symmetrical concatenation in the side branch.
- *FPHBN* [22]: The feature pyramid and hierarchical boosting network (FPHBN) is a recently proposed framework for crack detection and constructed on the main structure of HED.
- *FCN* [18]: A simplified hourglass shape network for crack detection with only 6 blocks.

- *U-Net* [23]: An U-shape auto-encoder network with scale-invariant merging between outputs from the encoder and the decoder.

In the second experiment, the methods to compare with are listed as follows:

- *ITTT* [48]: An iterative thresholding method controlled by the distance between current and previous thresholds.
- *CAT* [49]: A modified Otsu's thresholding method [37] with the enhancement of contrast via resizing the histogram.

VI. RESULTS AND DISCUSSION

A. COMPARISON WITH RESULTS FROM DIFFERENT FRAMEWORKS

The visual results of those DCNN frameworks for crack detection are depicted in Fig. 5, wherein autoencoder models such as SegNet, FPBHN, DeepCrack, and HCNNFP are capable of resisting the interference caused by texture, painted boundaries, and uneven lighting conditions. Notably, in addition effectively preserving completed contour of cracks better than other methods, including DeepCrack and FPHBN, the proposed HCNNFP is also able to remove those confusing patterns, as shown in the last five columns for DCD-7 dataset. The complexity of crack-like patterns actively contributes to a better F-measure with our method. Besides, although the results of DeepCrack and HCNNFP reach a similar level of sophistication, HCNNFP outputs a thinner outline of cracks. All these lead to less false-positive labeling around the contour and more accurate prediction. This advantage can be confirmed quantitatively by the measures AF_β and $MAPE$ as shown in the charts of Fig. 6 and 7.

For further comparison in the first experiment, the average measures for 6 DCNN approaches are listed in Table II and Table III. It can be seen that our HCNNFP obtains the highest AF_β and the lowest $MAPE$ for CrackForest, SydCrack, DCD, and GAPs datasets, and performs as the second-best in Crack500. The processing time of all the compared models is demonstrated in Table IV. Since all the tested networks are fed with the same data for testing and data loader, the difference of processing time can be considered as a relative comparison of computational consumption. As shown in Table IV, among the top three approaches in terms of crack detection accuracy, namely HCNNFP, DeepCrack, and FPHBN, our proposed approach ranked second in computational efficiency. More importantly, its ability to process cropped images at approximately 60 frames per second has demonstrated the capability of our method in real-time application. It is noted that the augmentation method used for SYDcrack and CrackForest is by cropping rather than resizing as in Crack500. Since resizing can generally weaken the representation of features with fewer details, DCD and GAPs are used here in the original size from the source provided by [44] with a higher fidelity level.

The feature preserving capability and high performance in crack detection as evaluated by those measures indicate the effectiveness of our approach overall. It is also worth noting that the top three models are all U-shape autoencoder while the fourth is also based on the first model architecture. This indicates the advantage of the proposed feature preservation branch applied to existing hierarchical architectures for vision-based monitoring. Specifically, recent autoencoder models such as DeepCrack, FPHBN and our HCNNFP performs better than the prototype autoencoders like SegNet. The main difference between them is that those updated models has an independent branch to integrate the output from different scales into a unified scale after resizing and refilling. This branch can be the key to the improvement of accuracy. Notably, the robustness of our proposed method over the range of interest for value β^2. Indeed, the relationship of F_β versus β^2 for the CrackForest dataset is shown in Fig. 9, where it can be seen that the fitness F_β of HCNNFP remains the highest for $\beta^2 \leq 1$ as compared to other existing deep learning techniques. In particular, the evaluation using the standard F-measure can also be seen in Fig. 9, where, at the point $\beta = 1$, the proposed HCNNFP gives the maximal value at around 0.88. This merit is also preserved if taking the arithmetic mean of F_β for the five datasets.

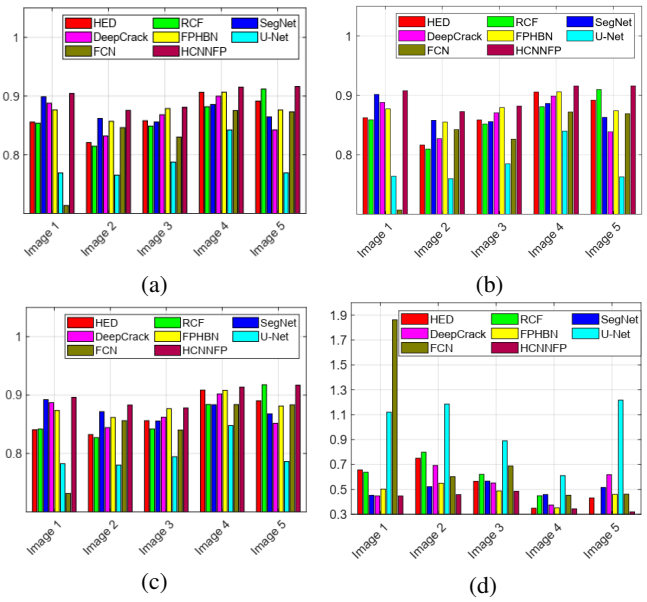

FIGURE 6: Quantitative results of the first five samples on crack images: (a) $F_\beta|\beta^2 = 0.25$, (b) $F_\beta|\beta^2 = 0.3$, (c) AF_β, (d) $MAPE$.

B. COMPARISON IN POST-PROCESSING

The binarization results are shown in Fig. 8. It can be seen that all the approaches can provide a high level of fitness to the ground truth. However, among them, CBAT presents a prediction map with the fewest crack labels. Moreover, as shown in the second row, although both thresholding methods are misled by the trace of insignificant dents, our CBAT can reduce the error by removing some false-positive pixels, and thus enhancing the precision rate. This improvement

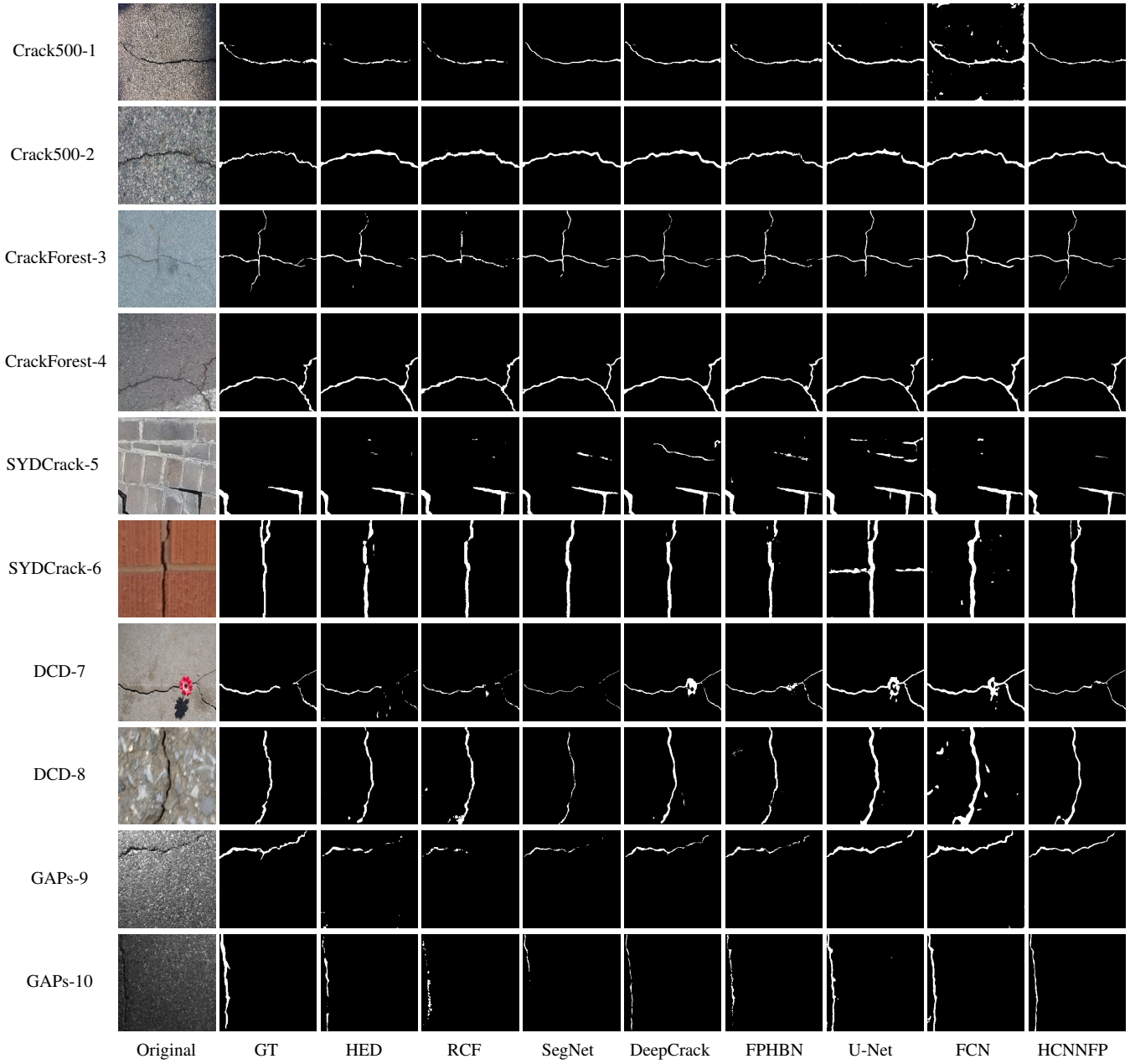

FIGURE 5: Detection results by six DCNN approaches with even threshold $T = 0.5$.

Methods	Crack500		CrackForest		SYDCrack		DCD		GAPs	
	β^2=0.25	β^2=0.3	β^2=0.25	β^2=0.3	β^2=0.25	β^2=0.3	β^2=0.25	β^2=0.3	β^2=0.25	β^2=0.3
HED	0.7877	0.7868	0.8699	0.8697	0.8414	0.8417	0.8350	0.8327	0.6664	0.6626
RCF	0.8034	0.8024	0.8612	0.8602	0.8462	0.8473	0.8542	0.8509	0.6568	0.6515
SegNet	0.8024	0.8021	0.8654	0.8642	0.8507	0.8509	0.8588	0.8570	0.7536	0.7530
DeepCrack	0.8068	0.8083	0.8765	0.8765	0.8514	0.8514	0.8667	0.8656	0.7642	0.7657
FPHBN	**0.8211**	**0.8207**	0.8773	0.8771	0.8510	0.8517	0.8671	0.8653	0.7890	0.7816
U-Net	0.7510	0.7542	0.7839	0.7853	0.7730	0.7778	0.7849	0.7897	0.6770	0.6831
FCN	0.8030	0.8042	0.8417	0.8452	0.8243	0.8274	0.8005	0.8049	0.7562	0.7619
HCNNFP	0.8179	0.8181	**0.8805**	**0.8797**	**0.8552**	**0.8551**	**0.8700**	**0.8688**	**0.7949**	**0.7933**

TABLE II: Comparison of F-measure F_β among eight DCNN approaches on five datasets.

is explained by the high credibility of CBAT in favor of crack features, and by its low credibility in shadow dents due to fewer rounds in the repetition of similar patterns in the training set. As such patterns are excluded by using CBAT, the prediction is closer to the ground truth. This has resulted in the highest values of AF_β and the lowest MAE

Methods	Crack500		CrackForest		SYDCrack		DCD		GAPs	
	AF_β	$MAPE$	AF_β	$MAPE$	AF_β	$MAPE$	AF_β	$MAPE$	AF_β	$MAPE$
HED	0.7849	0.9038	0.8692	0.5095	0.8432	0.7358	0.8277	0.9856	0.6542	1.2491
RCF	0.8005	0.8325	0.8579	0.5498	0.8508	0.7360	0.8434	0.9558	0.6511	1.1798
SegNet	0.8016	0.9619	0.8616	0.5430	0.8523	0.7156	0.8505	0.8602	0.7322	1.2090
DeepCrack	0.8122	0.8946	0.8767	0.4870	0.8522	0.6994	0.8632	0.7649	0.7538	0.9617
FPHBN	**0.8202**	**0.7782**	0.8768	0.4843	0.8542	0.6863	0.8615	0.7604	0.7662	0.8723
U-Net	0.7629	1.3783	0.7887	0.8330	0.7907	1.2924	0.8062	1.6089	0.6927	1.8614
FCN	0.8076	0.9251	0.8542	0.6225	0.8358	0.8500	0.8170	1.4950	0.7685	1.1841
HCNNFP	0.8188	0.8081	**0.8780**	**0.4807**	**0.8558**	**0.6725**	**0.8662**	**0.7520**	**0.7807**	**0.8503**

TABLE III: Comparison of average measures among eight DCNN approaches on five datasets.

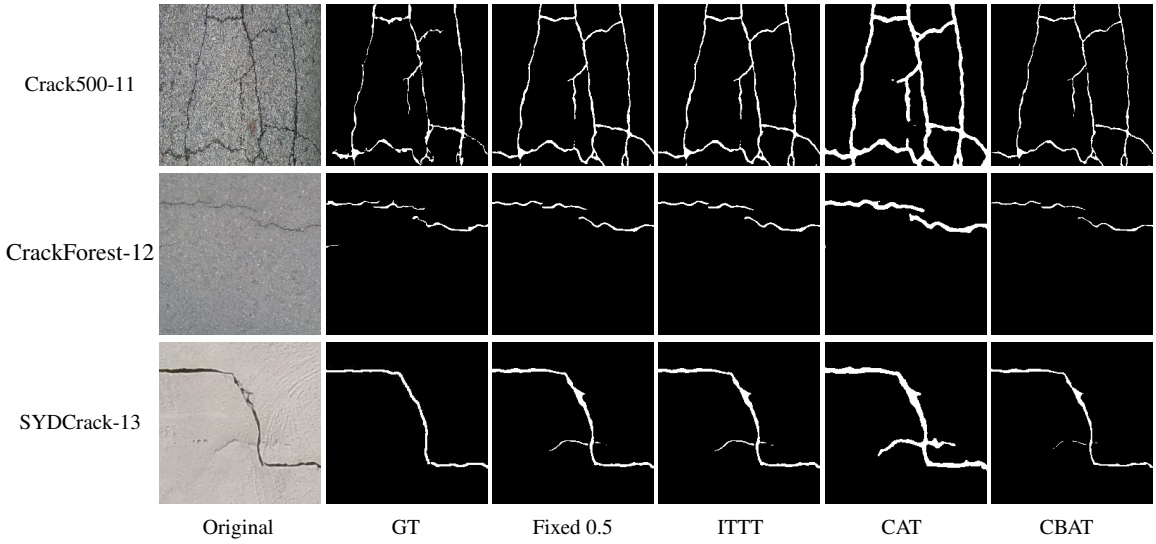

FIGURE 8: Binarization results of the probability map.

Methods	HED	RCF	SegNet	DeepCrack	FPHBN	U-Net	FCN	HCNNFP
Processing time(ms)	7.44	9.47	14.49	15.24	18.15	15.12	12.71	16.01

TABLE IV: Comparison of the processing time among eight DCNN approaches.

as shown in the chart of Fig. 10. The quantitative results of the four binarization approaches are presented in Table V and Table VI for the three data sets CrackForest, SYDCrack and Crack500. Among the tests on three datasets, almost all metrics are better after applying CBAT to the post-processing step.

Thresholds	Crack500		CrackForest		SYDCrack	
	β=0.25	β=0.3	β=0.25	β=0.3	β=0.25	β=0.3
Fixed 0.5	0.8179	0.8181	0.8805	0.8797	0.8552	0.8551
ITTT	0.8167	0.8171	0.8789	0.8785	0.8541	0.8543
CAT	0.7456	0.7506	0.7777	0.7837	0.8007	0.8054
CBAT	**0.8279**	**0.8258**	**0.8865**	**0.8858**	**0.8643**	**0.8619**

TABLE V: Comparison of F-measure F_β among binarization approaches: thresholding with fixed T=0.5, ITTT, CAT and CBAT.

Notably, for an ablation analysis, in addition to the comparison of DeepCrack and HCNNFP for the cases without and with our feature preserving branch as shown in Fig. 5 in the case of even binarization, the effect of autotuned thresholding is also presented in this comparison with our proposed CBAT. Indeed, as indicated in Table III, the AF_β are improved on all the datasets, especially on GAPs, with

Thresholds	Crack500		CrackForest		SYDCrack	
	AF_β	$MAPE$	AF_β	$MAPE$	AF_β	$MAPE$
Fixed 0.5	0.8188	0.8081	0.8780	0.4807	0.8558	0.6725
ITTT	0.8183	0.8188	0.8777	0.4855	0.8554	0.6790
CAT	0.7642	1.6535	0.8001	1.1365	0.8181	1.1509
CBAT	**0.8211**	**0.7431**	**0.8836**	**0.4757**	**0.8569**	**0.6366**

TABLE VI: Comparison of average measures among binarization approaches: thresholding with fixed T=0.5, ITTT, CAT and CBAT

an increase of 2.69%, while $MAPE$ drops by 11.14%. With the proposed feature preserving branch, more false-negative samples are rectified due to its robust mechanism in dealing with the nonlinearity. Also, a similar improvement can be seen in the comparison between our autotuned thresholding and raw binarization as quantified in Table VI. Those results verify the effectiveness and robustness of the proposed HCNNFP with feature preserving and autotuned thresholding.

C. DISCUSSION

Experimental results have demonstrated performance enhancements from the proposed hierarchical convolutional

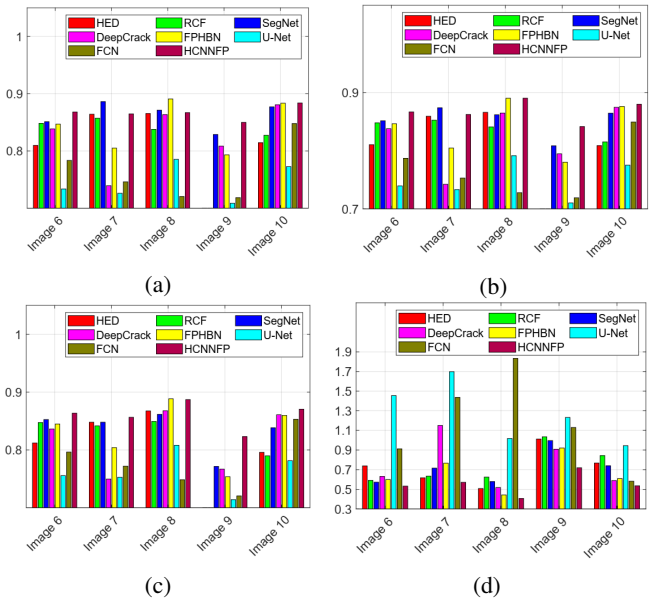

FIGURE 7: Quantitative results of the last five samples on crack images: (a) $F_\beta|\beta^2 = 0.25$, (b) $F_\beta|\beta^2 = 0.3$, (c) AF_β, (d) $MAPE$.

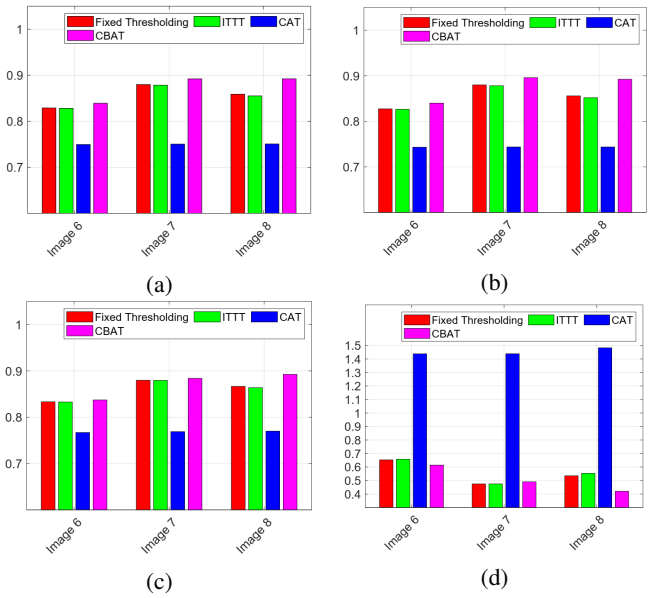

FIGURE 10: Quantitative results of fixed thresholding and CBAT on crack images: (a) $F_\beta|\beta^2 = 0.25$, (b) $F_\beta|\beta^2 = 0.3$, (c) AF_β, (d) $MAPE$.

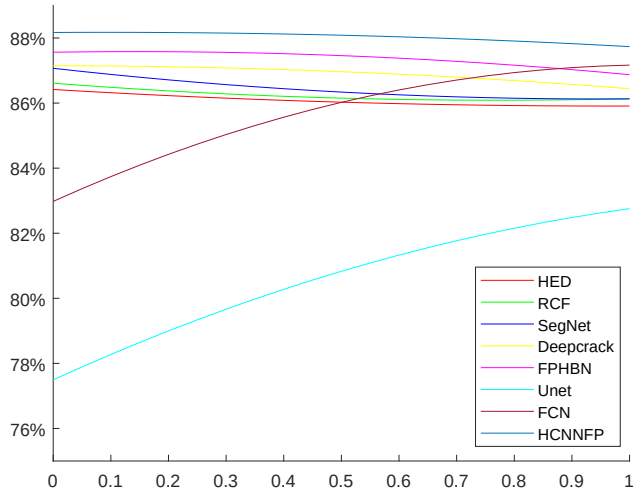

FIGURE 9: Distribution of F_β with respect to β^2

neural network with feature preserving for crack detection towards intelligent monitoring of transportation infrastructure. In the post-processing stage, the proposed intercontrast iterative thresholding also significantly contributes to improving binarization results for accurate feature extraction. Experimental results in crack detection on different datasets have shown the influences of redundant nonlinearity on the level of detail abstraction and the need for high credibility and scalability for reliable assessment of the surface defects and its attributes. These issues can be effectively dealt with by using the proposed feature preserving branch and intercontrast iterative thresholding algorithm. Moreover, errors in vision-based defect detection are often not fully reflected by the currently used evaluation with F_β. As indicated in Fig. 5, the shift in evaluation results is rather small where some parts of features may be missing. Here, a more comprehensive metric like the average F-measure AF_β offers a complementary criterion to consider also the effect of mislabeling due to unmatched labels. Future work will look at the incorporation of more information in post-processing with geometric filters to accommodate different shapes when classifying the probability with high credibility.

VII. CONCLUSION

This paper has presented a hybrid framework for detection of surface cracks in roads, tunnels or bridges. The proposed hierarchical convolutional neural network is equipped with a feature preserving branch to deal with the trade-off between nonlinearity and information loss. Moreover, the credibility of the features at the network output is further improved with a new intercontrast iterative algorithm based on Otsu thresholding to increase the detection accuracy. From the raw prediction, our enhanced hierarchical neural network can alleviate deviations caused by nonlinearity accumulated along with the network depth such that the upper-layer features become more linear and worth more weighting in labeling. At the post-processing stage, the contrast-based iterative thresholding can automatically search for a suitable boundary value in the probability map for accurate binarization, subject to a robust AF_β over a range of weighting between prediction and recall. As a result, the developed framework can successfully detect surface cracks of five different datasets for a road, a pavement, and a bridge subject to various texture levels. Extensive comparisons with the existing state-of-the-art deep learning convolutional neural networks for crack detection

has demonstrated the merits of the proposed approach.

REFERENCES

[1] W. Cao, Q. Liu, and Z. He, "Review of pavement defect detection methods," *IEEE Access*, vol. 8, pp. 14 531–14 544, 2020.

[2] Q. Song, Y. Wu, X. Xin, L. Yang, M. Yang, H. Chen, C. Liu, M. Hu, X. Chai, and J. Li, "Real-time tunnel crack analysis system via deep learning," *IEEE Access*, vol. 7, pp. 64 186–64 197, 2019.

[3] H. Li, D. Song, Y. Liu, and B. Li, "Automatic pavement crack detection by multi-scale image fusion," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2025–2036, 2018.

[4] T. H. Dinh, M. D. Phung, and Q. P. Ha, "Summit navigator: A novel approach for local maxima extraction," *IEEE Transactions on Image Processing*, vol. 29, pp. 551–564, 2020.

[5] N.-D. Hoang and Q.-L. Nguyen, "A novel method for asphalt pavement crack classification based on image processing and machine learning," *Engineering with Computers*, vol. 35, no. 2, pp. 487–498, 2019.

[6] W. Xu, Z. Tang, J. Zhou, and J. Ding, "Pavement crack detection based on saliency and statistical features," in *20th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2013, pp. 4093–4097.

[7] V. Kaul, A. Yezzi, and Y. Tsai, "Detecting curves with unknown endpoints and arbitrary topology using minimal paths," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 10, pp. 1952–1965, 2011.

[8] D. Ai, G. Jiang, L. S. Kei, and C. Li, "Automatic pixel-level pavement crack detection using information of multi-scale neighborhoods," *IEEE Access*, vol. 6, pp. 24 452–24 463, 2018.

[9] Y. Shi, L. Cui, Z. Qi, F. Meng, and Z. Chen, "Automatic road crack detection using random structured forests," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 12, pp. 3434–3445, 2016.

[10] L. Zhang, G. Zhou, Y. Han, H. Lin, and Y. Wu, "Application of internet of things technology and convolutional neural network model in bridge crack detection," *IEEE Access*, vol. 6, pp. 39 442–39 451, 2018.

[11] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[12] Q. Zhu, M. Duong, and Q. Ha, "Crack detection using enhanced hierarchical convolutional neural networks," in *Australasian Conference on Robotics and Automation (ACRA)*. AARA, 2019, pp. 1–8.

[13] W. Wang and C. Su, "Convolutional neural network-based pavement crack segmentation using pyramid attention network," *IEEE Access*, vol. 8, pp. 206 548–206 558, 2020.

[14] Q. Zou, Z. Zhang, Q. Li, X. Qi, Q. Wang, and S. Wang, "Deepcrack: Learning hierarchical convolutional features for crack detection," *IEEE Transactions on Image Processing*, vol. 28, no. 3, pp. 1498–1512, 2018.

[15] V. T. Hoang, M. D. Phung, T. H. Dinh, and Q. P. Ha, "System architecture for real-time surface inspection using multiple uavs," *IEEE Systems Journal*, pp. 1–12, 2019.

[16] L. Zhang, F. Yang, Y. D. Zhang, and Y. J. Zhu, "Road crack detection using deep convolutional neural network," in *2016 IEEE international conference on image processing (ICIP)*. IEEE, 2016, pp. 3708–3712.

[17] N.-D. Hoang, Q.-L. Nguyen, and V.-D. Tran, "Automatic recognition of asphalt pavement cracks using metaheuristic optimized edge detection algorithms and convolution neural network," *Automation in Construction*, vol. 94, pp. 203–213, 2018.

[18] C. V. Dung and L. D. Anh, "Autonomous concrete crack detection using deep fully convolutional neural network," *Automation in Construction*, vol. 99, pp. 52–58, 2019.

[19] A. Zhang, K. C. Wang, B. Li, E. Yang, X. Dai, Y. Peng, Y. Fei, Y. Liu, J. Q. Li, and C. Chen, "Automated pixel-level pavement crack detection on 3d asphalt surfaces using a deep-learning network," *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 10, pp. 805–819, 2017.

[20] Y. Fei, K. C. Wang, A. Zhang, C. Chen, J. Q. Li, Y. Liu, G. Yang, and B. Li, "Pixel-level cracking detection on 3d asphalt pavement images through deep-learning-based cracknet-v," *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[21] Q. Mei, M. Gül, and M. R. Azim, "Densely connected deep neural network considering connectivity of pixels for automatic crack detection," *Automation in Construction*, vol. 110, p. 103018, 2020.

[22] F. Yang, L. Zhang, S. Yu, D. Prokhorov, X. Mei, and H. Ling, "Feature pyramid and hierarchical boosting network for pavement crack detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 4, pp. 1525–1535, 2020.

[23] Z. Liu, Y. Cao, Y. Wang, and W. Wang, "Computer vision-based concrete crack detection using u-net fully convolutional networks," *Automation in Construction*, vol. 104, pp. 129–139, 2019.

[24] F.-C. Chen and M. R. Jahanshahi, "Nb-cnn: Deep learning-based crack detection using convolutional neural network and naïve bayes data fusion," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 5, pp. 4392–4400, 2017.

[25] H. Chen, Q. Hu, B. Zhai, H. Chen, and K. Liu, "A robust weakly supervised learning of deep conv-nets for surface defect inspection," *Neural Computing and Applications*, pp. 1–16, 2020.

[26] H. Dong, K. Song, Y. He, J. Xu, Y. Yan, and Q. Meng, "Pga-net: Pyramid feature fusion and global context attention network for automated surface defect detection," *IEEE Transactions on Industrial Informatics*, 2019.

[27] W. Choi and Y.-J. Cha, "Sddnet: Real-time crack segmentation," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 9, pp. 8016–8025, 2019.

[28] H. Wang and D.-Y. Yeung, "Towards bayesian deep learning: A framework and some existing methods," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 12, pp. 3395–3408, 2016.

[29] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[30] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[31] A. Borji, M.-M. Cheng, H. Jiang, and J. Li, "Salient object detection: A benchmark," *IEEE transactions on image processing*, vol. 24, no. 12, pp. 5706–5722, 2015.

[32] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 658–666.

[33] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, and S.-M. Hu, "Global contrast based salient region detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 569–582, 2014.

[34] A. Milan, T. Pham, K. Vijay, D. Morrison, A. W. Tow, L. Liu, J. Erskine, R. Grinover, A. Gurman, T. Hunn *et al.*, "Semantic segmentation from limited training data," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1908–1915.

[35] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. H. Torr, "Deeply supervised salient object detection with short connections," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3203–3212.

[36] A. De Myttenaere, B. Golden, B. Le Grand, and F. Rossi, "Mean absolute percentage error for regression models," *Neurocomputing*, vol. 192, pp. 38–48, 2016.

[37] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[38] M. D. Levine and A. M. Nazif, "Dynamic measurement of computer generated image segmentations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 2, pp. 155–164, 1985.

[39] V. T. Hoang, M. D. Phung, T. H. Dinh, Q. Zhu, and Q. Ha, "Reconfigurable multi-uav formation using angle-encoded pso," in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2019, pp. 1670–1675.

[40] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1026–1034.

[41] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015, arXiv preprint arXiv:1412.6980.

[42] Y. Liu, J. Yao, X. Lu, R. Xie, and L. Li, "Deepcrack: A deep hierarchical feature learning architecture for crack segmentation," *Neurocomputing*, vol. 338, pp. 139–153, 2019.

[43] M. Eisenbach, R. Stricker, D. Seichter, K. Amende, K. Debes, M. Sesselmann, D. Ebersbach, U. Stoeckert, and H.-M. Gross, "How to get pavement distress detection ready for deep learning? a systematic approach," in *2017 international joint conference on neural networks (IJCNN)*. IEEE, 2017, pp. 2039–2047.

[44] K. Liu, X. Han, and B. M. Chen, "Deep learning based automatic crack detection and segmentation for unmanned aerial vehicle inspections,"

in *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2019, pp. 381–387.
[45] H. Pan, Z. Pang, Y. Wang, Y. Wang, and L. Chen, "A new image recognition and classification method combining transfer learning algorithm and mobilenet model for welding defects," *IEEE Access*, vol. 8, pp. 119 951–119 960, 2020.
[46] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICIP)*, 2015, pp. 1395–1403.
[47] Y. Liu, M.-M. Cheng, X. Hu, K. Wang, and X. Bai, "Richer convolutional features for edge detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3000–3009.
[48] H. Cai, Z. Yang, X. Cao, W. Xia, and X. Xu, "A new iterative triclass thresholding technique in image segmentation," *IEEE Transactions on Image Processing*, vol. 23, no. 3, pp. 1038–1046, 2014.
[49] M. Win, A. Bushroa, M. Hassan, N. Hilman, and A. Ide-Ektessabi, "A contrast adjustment thresholding method for surface defect detection based on mesoscopy," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 3, pp. 642–649, 2015.

QIUCHEN ZHU received the M.Eng. degree from Huazhong University of Science and Technology, Wuhan, China, in 2017.

He is currently pursuing the Ph.D. degree with the School of Electrical and Data Engineering, University of Technology Sydney, Australia. His research interests include machine vision, image processing, probabilistic representation and uncertainty of deep learning.

TRAN HIEP DINH received his M.Sc. degree in mechatronics from the Leibniz University Hanover, Germany, and PhD degree in engineering from the University of Technology Sydney, Australia, in 2010 and 2020, respectively.

He is currently with the Faculty of Engineering Mechanics and Automation, VNU University of Engineering and Technology. His research interests include image processing, robotics, and machine learning.

MANH DUONG PHUNG received the B.Sc. and Ph.D. degrees from Vietnam National University, Hanoi, Vietnam in 2005 and 2015, respectively.

He is currently a lecturer at University of Technology Sydney and Vietnam National University, Hanoi. He has conducted a number of research projects with industry and international partners such as Eye tracking with NTT Cyber Solution Laboratory, Japan, Telehealth with Mechatronics and Automation Laboratory of National University of Singapore, 3-D hand tracking with Samsung Vietnam Mobile R& D Centre, and Robotics and Automation in Construction with Department of Defence, Australia. His research interests include automation in construction, unmanned aerial vehicles, mobile robot localization and mapping, and optimization.

Q. P. HA (SM'13) received the B.E. degree fro Ho Chi Minh City University of Technology, Ho Chi Minh City, Vietnam, in electrical engineering in 1983 and the Ph.D. degrees from Moscow Power Engineering Institute, Moscow, Russia, in complex systems and control in 1993, and the University of Tasmania, Australia, in intelligent systems, in 1997.

He is currently an Associate Professor with the School of Electrical and Data Engineering of the Faculty of Engineering and Information Technology, University of Technology, Sydney, Australia. His research interests include automation, robotics, and control systems. Dr. Ha has been on the Board of Directors of the International Association of Automation and Robotics in Construction since 2007. He was Conference Chair/Co-Chair of several international conferences on automation and intelligent systems.

He has been on the editorial board of the IEEE Transactions on Automation Science and Engineering (2009–2013), Automation in Construction, Robotica, Electronics, and some others. He was the recipient of a number of best paper awards from the IEEE, IAARC, and Engineers Australia, including the Sir George Julius Medal in 2015.