A Bidirectional Self-Rectifying Network with Bayesian Modeling for Vision-Based Crack Detection

Qiuchen Zhu, *Student Member, IEEE*, Quang Ha, *Senior Member, IEEE*

***Abstract*— Robotic vision is increasingly applied for surface inspection of built infrastructure. For this, it is essential to develop robust algorithms for semantic segmentation. This paper presents a deep learning approach using a bidirectional self-rectifying network with Bayesian modeling (BSNBM) for improving detection accuracy, in dealing with the embedded uncertainty caused by false-positive labels and nonlinearity in sequentially-convolutional blocks. For integration with residual encoders, a feature preserving branch is designed, wherein the output of previous dilated convolutional blocks (DCBs) is upsizedly or downsizedly passed on and concatenated with the following blocks recursively and bidirectionally. Further, to achieve robustness in feature representation with an acceptable level of credibility, convolutional kernels are randomized via a Bayesian model and adjusted per evidence update. As such, the network becomes less sensitive to uncertainty and redundant nonlinearity, which are inevitable in activation layers. Experimental results confirm the advantage of our BSNBM over current crack detection approaches.**

***Index Terms*— Deep learning, Bayesian inference, hierarchical convolutional neural network, crack detection.**

I. Introduction

Surface crack is a structural health indicator for the defect in transportation and civil assets such as roads, bridges and buildings. Their effective maintenance and evacuation plans in residential zones for safety rest with the prompt inspection to such signs of abnormal cracks. For dangerous and unattainable inspection sites, it is wise to conduct robotic monitoring on cracks using e.g., unmanned aerial vehicles (UAVs), unmanned ground vehicles (UGVs). Vision-based crack detection requires a strong technique to address the important issue of similar binary segmentation [1] from the extraction of generic features. With advances in neural computing, deep convolutional neural networks (DCNN) have emerged as a powerful tool that can effectively address this requirement in order to improve robustness and accuracy by learning from data.

This work was supported by China Scholarship Council.

Qiuchen Zhu, is now with the School of Electrical and Data Engineering, Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, 2007 NSW, Australia (e-mail: Qiuchen.Zhu@student.uts.edu.au).

Quang Ha, is now with the School of Electrical and Data Engineering, Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, 2007 NSW, Australia (e-mail: Quang.Ha@uts.edu.au).

In deep learning, hierarchical connections have been proved to be a network structure with stage-like transitions suitable for categorical distinctions in classification. Such a structure can better preserve the detail of patterns due to its multiscale feature abstraction and superposed output, as required for improving the performance of an image processing system used in crack detection [2]. In addition, a merging residual module [3] inspired by the residual effect of biological eyes also extends the capacity of networks in terms of pattern abstraction. The residual modules equipped at the downsampling pipeline can contribute positively to remedy the effect of blurry images [4].

As with human vision, object recognition of occluded objects would require recurrent processing with lateral connectivity [5]. In this regard, recurrent neural networks (RNN), originally designed to exploit the context in natural language processing, have been applied to recent DCNN models to improve the graphical abstractions. The latent correlation between hierarchical abstractions enables DCNNs to track the missing detail using the context information obtained along the processing pipeline [6]. Despite the limited abstractions provided by DCNNs, a similar self-rectification as recurrent modeling can still be achieved by using iterative superposition [7]. In addition, dilated convolution is commonly employed in DCNNs to replace the standard convolutional layer. As pointed out by [8], the effective receptive field of the dilated convolution is larger than the standard one. Hence, the dilated convolutional layers can provide enough receipt fields with fewer layers and parameters to form a simplified architecture. Incorporating those improvements, a hierarchically structured residual network is proposed in this work for crack detection.

From the probabilistic perspective, the output of a neural network can be regarded as the statistic representation of data distribution. In most applications, deep learning architectures mainly focus on the hypothesis of frequentist statistics, which provides promising results after training from a large dataset. However, frequentist networks with fixed parameters often become overconfident on uncertainties and false-negative samples such as misleading marks. These could affect the network's robustness with respect to a diversity of images obtained from unmanned vehicles. For example, the training set may be obtained from images with a better light condition than the test images. In this case, the model is likely to make confident predictions on crack candidates even when the test images themselves contain vagueness with a low

resolution. On the other hand, those networks with Bayesian modeling [9] can provide an alternative to the pipeline for inference. In Bayesian modeling, parameters of a model no longer have fixed values but are generally determined and optimized by its evidence [10]. Such Bayesian models possess an advantage in achieving more accurate inference for crack detection. However, the naïve Bayesian model used in vision-based inspection is often mathematically intractable in real-world data. To address this problem, a reasonable solution is to find an alternative distribution in lieu of the original model for inference [11].

In this paper, Bayesian modeling is incorporated in the proposed directional self-rectifying network to identify surface cracks on a monorail bridge by robotic inspection with UAVs. The images, collected by a formation of UAVs, are initially converted to a probability map of crack candidates via hierarchical feature extraction and Bayesian inference. The crack patterns are then binarily restructured based on the probability of the crack. Experiments on different datasets and on images captured by our airborne robots verify the effectiveness of the proposed bidirectional self-rectifying network with Bayesian modeling (BSNBM) in comparison with recent crack detection approaches available in the existing literature.

Our contributions include: (i) to propose a new hierarchical network to visually detect surface cracks, whereby the abstractions of the image are bidirectionally extracted to result in a balanced prediction, (ii) to develop lateral recurrent connections between layers to exploit the context information of correlated abstractions for feature preservation, and (iii) to develop a gradient-sharing approach to embed Bayesian inference in the network for uncertainty handling, wherein the kernel parameters are updated under the supervision of a hybrid loss.

The paper is organized as follows. After the introduction, Section II presents the architecture of the proposed framework and its probabilistic modeling for crack identification. In Section III the Bayesian inference mechanism is developed and applied to the proposed framework. Section IV describes our robotic system for image acquisition, and the implementation detail of two experiments for verification of the BSNBM merit followed by a comparison with the relevant deep learning framework for pixel-wise segmentation. Section V provides the experimental results along with an ablation study. Finally, a conclusion is drawn in Section VI.

II. Hierarchical Residual Convolutional Neural Network

A. Network Architecture

The structure of the proposed neural network is shown in Fig. 1. Unlike the classic end-to-end autoencoder model [12] with 5 convolutional blocks in the encoder side, our main network here consists of three dilated convolutional blocks (DCBs) with one dilated convolutional layer [13] sandwiched by two standard convolutional layers per block. Those blocks preserve the hierarchical abstractions of features in 3 different scales. Inside the k^{th} block, each layer conducts a standard or dilated size-invariant convolutional operation with a preset dilation rate (DR) and a quarter-size max-pooling operation for the abstraction of crack features with $2^{(5+k)}$ channels. With the combination of several convolutional layers, the feature map is refined after each block. Compared to the standard ones, dilated convolutional layers process the image tensor using sparser kernels with the same parameters as the original, which extends the receipt field of the signal layer. In this case, the DCBs can cover adequately reception with fewer layers and parameters than the encoder of SegNet [12]. The convolutional abstraction from each DCB is further enhanced by a forward and a reverse enhancement branch (F/REB) bidirectionally and finally fed into a feature merging net to produce the final probability map of the crack using multiscale fusion.

B. Information Loss

We first consider the information loss in association of an image feature. For this, let denote a training sample as $D = \{(Y|X)\}$, where $X = \{x_{ij}\}$ and $Y = \{y_{ij}\}$ respectively represent the pixel values of the original image and its corresponding binary ground-truth mask, both of dimension $I \times J$, and correspondingly an indicator defined as,

$$\mathbf{1}_{y_{ij}}(x_{ij}) = \begin{cases} 1 & \text{if } x \to (y = 1) \\ 0 & \text{if } x \to (y = 0). \end{cases} \tag{1}$$

The probability of an arbitrary pixel x_{ij} being the landmark can be calculated from the Boltzmann distribution $\phi(x_{ij}, y_{ij})$ as,

$$P(y_{ij}|x_{ij}) = \frac{\phi(x_{ij}, y_{ij} = 1)}{\phi(x_{ij}, y_{ij} = 1) + \phi(x_{ij}, y_{ij} = 0)} = \frac{1}{1 + e^{-x_{ij}}}. \tag{2}$$

Now for an arbitrary feature f_{ij} in the feature map , the pixel-wise probability $P(f_{ij})$ is obtained similarly as [4],

$$p(f_{ij}) = \frac{1}{1 + e^{-f_{ij}}}. \tag{3}$$

The associated information loss for feature f_{ij}^k at the k^{th} block can be expressed via its binary entropy as,

$$l(f_{ij}^k) = -y_{ij} \ln(p(f_{ij}^k)) - (1 - y_{ij}) \ln(1 - p(f_{ij}^k)). \tag{4}$$

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 losses $l(f_{ij}^{fused})$ in the fused map $F^{fused} = \{f_{ij}^{fused}\}$ for all pixels, i.e.

$$\mathcal{L}_f = \sum_{i=1}^{I} \sum_{j=1}^{J} \left(l(f_{ij}^{fused}) + \sum_{k=1}^{3} l(f_{ij}^k) \right). \tag{5}$$

As discussed in [4], due to the necessity of nonlinear activation in deep learning models, probability (3) on one hand may become inaccurate as long as the relationship between f_{ij} and the pixel values x_{ij} deviates from a linearly-dependence with the accumulation of the depth. On the other hand, nonlinearity also facilitates the complexity of representation that benefits

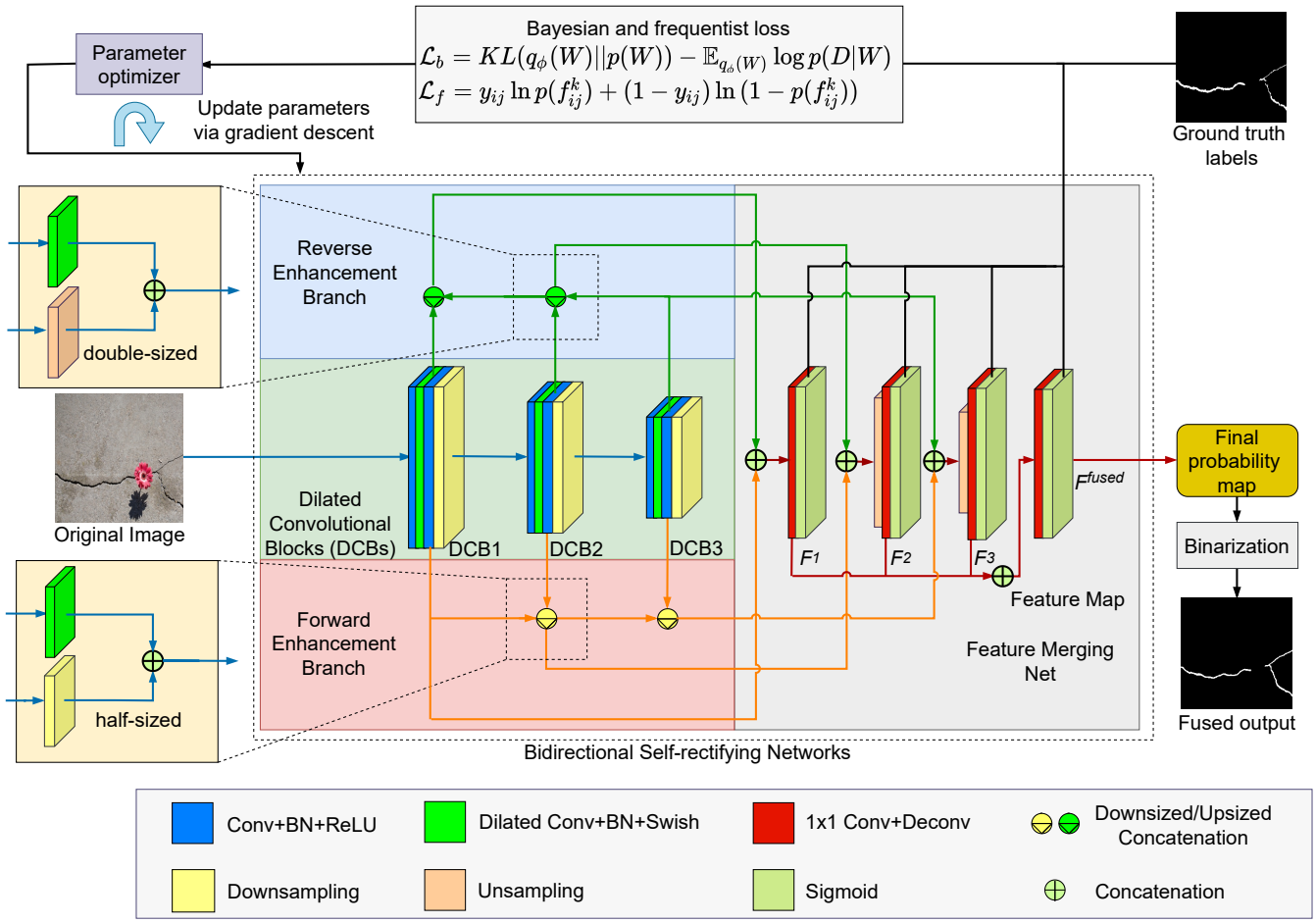

Fig. 1: Architecture for the bidirectional self-rectifying network with Bayesian modeling (BSNBM).

the diversity of the abstraction. This requires an effective technique to address this nonlinearity trade-off and to reduce the dependence of unilateral outputs, which is the focus of this paper.

C. Activation Layers and Statistical Dropout

To better determine the network output mathematically while reducing its nonlinearity as well as increasing generality, a recent activation function, the Swish [14], is used here instead of the commonly-used rectified linear unit (ReLU) [12]. Compared with ReLU, the Swish function is continuously differentiable as modified from the sigmoid activation function:

$$\text{swish}(x) = x \cdot \text{sigmoid}(x). \tag{6}$$

Consequently, the backpropagation via the Swish is more stable and effective in dealing with the issue of uncertainty in deep neural networks.

To improve network robustness, uncertainty associated with the whole pipeline needs to be adequately handled. For example, the sparser kernels of dilated convolution in DCBs may lower fidelity of the abstraction at the output, hence requiring an effective activation function. Moreover, uncertainty in the input-output relation $y = Wx$ can be modeled from the Bayesian perspective by a posterior distribution $p(W|x)$ with a random distribution $q(W)$. Such model can be approximated by the expectation of the distribution $q(W)$, or Monte Carlo dropout [15],

$$y = \mathbb{E}_{q(W)}(x), \tag{7}$$

which could be practically obtained by performing a number of stochastic forward passes through the network and then averaging the results. In this paper, the Swish actuation function along with the Bayesian model for uncertainty in non-parametric layers are utilized to positively contribute to enhanced network robustness with the proposed BSNBM.

D. Bidirectional Self-Rectifying Modules

In natural language processing, bidirectional recurrent networks [16] can effectively reduce the information loss due to the linkage of the context between adjacent hidden states. This technique is also applied to deep learning [17] for pattern reconstruction. However, it is quite different to implement a completed recurrent network for image processing due to the limited range of spectral contexts [18]. Alternatively, the RNN methodology can be integrated into DCNN by using limited recurrent units [7]. Here by considering the feature f_{ij}^k as a dynamic imaging state in lieu of an independent abstraction, we propose an opposite pair of feature preserving branches at the end of the DCBs to externally extend or shrink to adjust the abstraction weight from the following DCB along the routine.

To take into account the difference in scales, the concatenation here is conducted with a resized operation. As illustrated by the forward branch illustrated shown in Fig. 1, the width of the larger tensor shrinks to a half size using an additional downsampling process. Two half-size tensors are then merged and fed to the next layers with superimposed channels. Due to an increase in channel possession of the upper-layer features, they weigh more than the lower-layer ones in the abstraction of graphic information. This is opposite to the reverse branch, as depicted also in the network structure figure. With the proposed resizing to retain the number of network parameters, the forward-wise feature $f_{ij}^{F,k}$ can be formulated like the state function of recurrent networks as [16],

$$f_{ij}^{F,k} = \begin{cases} D_{ij}^{k} & k = 1 \\ D_{ij}^{k} + \mathbb{E}_{q(W)}(conv(f_{ij}^{F,k-1})), & k = 2,3 \end{cases} \tag{8}$$

where D_{ij}^{k} is the pixel output from the k^{th} DCB, and $conv$ is the convolutional operation. Similarly, the reverse-wise feature $f_{ij}^{R,k}$ can be computed by:

$$f_{ij}^{R,k} = \begin{cases} D_{ij}^{k} + \mathbb{E}_{q(W)}(conv(f_{ij}^{R,k+1})) & k = 1,2 \\ D_{ij}^{k} & k = 3. \end{cases} \tag{9}$$

Finally, at the output, the merged feature f_{ij}^{k} at the scale k is obtained as,

$$f_{ij}^{k} = f_{ij}^{F,k} + f_{ij}^{R,k} \quad (k = 1,2,3). \tag{10}$$

The designed structure is to improve the quality of feature extraction by balancing the effect of redundant nonlinearity and complexity of the representation in the network.

In this paper, by incorporating additionally a Bayesian objective function $\mathcal{L}_b$, the total loss of the proposed network now becomes:

$$\mathcal{L} = W_f \mathcal{L}_f + W_b \mathcal{L}_b. \tag{11}$$

where W_f and W_b are respectively the weights of the frequentist and Bayesian loss for normalization. The selection of W_f and W_b is discussed in the section on our experiments while the determination of $\mathcal{L}_b$ is presented in the next section.

III. Bayesian modeling

The architecture mentioned in the previous section presents a non-quantitative methodology towards the alleviation of uncertainty influence. Hence, the network performance can be improved by modification but not optimization as long as the uncertainty of the model is not quantified. Moreover, in the trainable convolution layers, discrete parameters are preserved after training, and hence every feature representation even the outliers shares the same level of credibility as with frequentist inference. This may finally lead to overconfidence in the prediction. As a remedy, the trainable modules in this paper are developed by using Bayesian inference [10] along with Kullback-Leibler (KL) optimization.

A. Bayesian Inference

Let us consider the model W as a distribution of likelihood rather than discrete values from the discussion above. In the trainable layers of the Bayesian model, such distribution W over evidence can be calculated by referring to the training set $\mathcal{D}$ as per the Bayes' theorem:

$$p\left(W \mid \mathcal{D}\right) = \frac{p\left(\mathcal{D} \mid W\right) p(W)}{\int p\left(\mathcal{D} \mid W\right) p(W) dW}. \tag{12}$$

For a new sample $\{x^*, y^*\}$, an unbiased estimation of the probability can be given by using Monte Carlo sampling [19] on the output of the network with M samples:

$$\mathbb{E}_{p(W|\mathcal{D})} p\left(y^* \mid x^*, W\right) \simeq \frac{1}{M} \sum_{m=1}^{M} p\left(y^* \mid x^*, W^{(m)}\right) \tag{13}$$

where $W^{(m)}$ is randomly picked from the conditional distribution of W. Therefore, the posterior probability $p\left(W \mid X, Y\right)$ remains the key to unbiased estimation of a target probability.

In practice, the conditional probability $p(W \mid \mathcal{D})$ can be commonly estimated by using an approximate function, which is a random distribution $q(W)$, i.e. $p(W \mid \mathcal{D}) \approx q(W)$.

B. Optimization Target

Bayesian inference often involves the optimization of an objective function to be properly formulated, for which a solution may not be available. A possible alternative for this is the Kullback-Leibler (KL) divergence, a measure in terms of the distance between distributions. Specifically for the approximation $q(W)$ of $p(W|\mathcal{D})$, the K-L divergence can be obtained as,

$$KL(q(W) \| p(W \mid \mathcal{D})) = \int q(W) \log \frac{q(W)}{p(W \mid \mathcal{D})} dW \geq 0. \tag{14}$$

To achieve the closest estimation of $p(W \mid \mathcal{D})$, the KL divergence should approach the minimum during the training.

It can be noted that the prior distribution of $\mathcal{D}$ can be expressed in terms of KL divergence as [20],

$$\begin{aligned} \log p(\mathcal{D}) &= \int q\left(W\right) \log p(\mathcal{D}) dW = \int q\left(W\right) \log \frac{p(\mathcal{D}, W)}{p(W \mid \mathcal{D})} dW \\ &= \int q(W) \log \frac{q(W)}{p(W \mid \mathcal{D})} dW + \int q(W) \log \frac{p(\mathcal{D}, W)}{q(W)} dW \\ &= KL(q(W) \| p(W \mid \mathcal{D})) + \mathcal{L}(q(W)) \geq \mathcal{L}(q(W)), \end{aligned} \tag{15}$$

where $\mathcal{L}(q(W))$ is the evidence lower bound (ELBO):

$$\mathcal{L}(q(W)) = \int q(W) \log \frac{p(\mathcal{D}, W)}{q(W)} dW. \tag{16}$$

For the closest estimation of KL divergence, our optimization target is then equivalent to maximizing the ELBO since $\log p(\mathcal{D})$ is a constant for a given dataset $\mathcal{D}$. Now from the expression of the ELBO (16), $\mathcal{L}(q(W))$ can be decomposed into two terms as,

$$\begin{aligned} \mathcal{L}(q(W)) = &\int q(W) \log p(\mathcal{D} \mid W) dW \\ &+ \int q(W) \log \frac{p(W)}{q(W)} dW. \end{aligned} \tag{17}$$

Alternatively, by using the K-L divergence, we obtain

$$\mathcal{L}(q(W)) = \mathbb{E}_{q(W)} \log p(\mathcal{D} \mid W) - KL(q(W) \| p(W)). \quad (18)$$

In practice, $q(W)$ is often designed to follow a Gaussian distribution $\emptyset$ of mean $\boldsymbol{\mu}$ and standard deviation $\boldsymbol{\sigma}$, i.e. $\emptyset \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$. The ELBO under this distribution is therefore,

$$\mathcal{L}\left(q_{\emptyset}(W)\right) = \mathbb{E}_{q_{\emptyset}(W)} \log p(\mathcal{D} \mid W) - KL\left(q_{\emptyset}(W) \| p(W)\right). \quad (19)$$

To utilize the gradient descent method in deep neural networks, the objective function of Bayesian inference here is to minimize the opposite of the ELBO $\mathcal{L}\left(q_{\emptyset}(W)\right)$. Such Bayesian objective function per sampling is expressed as [21],

$$\begin{aligned} \mathcal{L}_b &= -\mathcal{L}\left(q_{\emptyset}(W)\right) \\ &= KL\left(q_{\emptyset}(W) \| p(W)\right) - \mathbb{E}_{q_{\emptyset}(W)} \log p(\mathcal{D} \mid W) \\ &\approx \sum_{m=1}^{M} \log q_{\emptyset}\left(W^{(m)} \mid \mathcal{D}\right) - \log p\left(W^{(m)}\right) \\ &\quad - \log p\left(\mathcal{D} \mid W^{(m)}\right), \end{aligned} \quad (20)$$

where $q_{\emptyset}(W^{(m)})$ and $p(W^{(m)})$ are sampled respectively from the estimated Gaussian distribution and the prior distribution of W. This component can be preset or estimated before training. The other component, $p\left(\mathcal{D} \mid W^{(m)}\right)$, is the probabilistic value obtained at the end of the frequentist network.

Note that the weighting matrix W can also be reparameterized by an arbitrary distribution of an independent hyper parameter ϵ,

whereby the new weights in the same architecture can be recast as

$$W = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, I). \quad (21)$$

Here, $\boldsymbol{\epsilon}$ is commonly modeled as a Gaussian process and $\odot$ is the component-wise multiplication. Now, the expected partial derivative of ELBO can be calculated as [21],

$$\frac{\partial \mathbb{E}[\mathcal{L}(q_{\emptyset}(W))]}{\partial \emptyset} = \mathbb{E}\left[\frac{\partial \mathcal{L}(q_{\emptyset}(W))}{\partial W}\frac{\partial W}{\partial \emptyset} + \frac{\partial \mathcal{L}(q_{\emptyset}(W))}{\partial \emptyset}\right]. \quad (22)$$

Accordingly, gradients of the mean and standard deviation can be obtained respectively as,

$$\nabla_{\mu} = \frac{\partial \mathcal{L}(W, \emptyset)}{\partial W} + \frac{\partial \mathcal{L}(W, \theta)}{\partial \boldsymbol{\mu}}, \quad (23)$$

$$\nabla_{\boldsymbol{\sigma}} = \frac{\partial \mathcal{L}(W, \emptyset)}{\partial W}\boldsymbol{\epsilon} + \frac{\partial \mathcal{L}(W, \emptyset)}{\partial \boldsymbol{\sigma}}. \quad (24)$$

Those obtained parameters can be uploaded using the same routine of the backpropagation as in DCNN [22].

The pseudo-code for the proposed Bayesian inference using the gradient descent method with a learning rate $\alpha > 0$ is described in Algorithm 1. In accordance, the mean and the deviation of the distribution W are updated during the training process. Figure 2 illustrates the Bayesian inference in trainable convolutional kernels for the total loss function of the proposed network. Therein, W^m is the weight collected via the m^{th} sampling.

Algorithm 1 Bayesian estimation

initialize the mean of W randomly
repeat
  Sample $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$
  Calculate $W = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$
  Calculate $\mathcal{L}$
  Calculate the gradient of the mean $\nabla_{\boldsymbol{\mu}} = \frac{\partial \mathcal{L}(W,\emptyset)}{\partial W} + \frac{\partial \mathcal{L}(W,\emptyset)}{\partial \boldsymbol{\mu}}$
  Calculate the gradient of the standard deviation $\nabla_{\boldsymbol{\sigma}} = \frac{\partial \mathcal{L}(W,\emptyset)}{\partial W}\epsilon + \frac{\partial \mathcal{L}(W,\emptyset)}{\partial \boldsymbol{\sigma}}$
  $\boldsymbol{\mu} \leftarrow \boldsymbol{\mu} - \alpha \nabla_{\boldsymbol{\mu}}$
  $\boldsymbol{\sigma} \leftarrow \boldsymbol{\sigma} - \alpha \nabla_{\boldsymbol{\sigma}}$
until $\mathcal{L}$ converges

IV. Experiments and Evaluation

Here, parameters for training are obtained at first by using He Normal initialization [23]. The training is conducted at a learning rate of 1e-5 with 10-fold cross validation and optimized by Adaptive Moment Estimation (Adam) [24]. For the performance of the models used in the comparison, the maximum training epoch is set as 30. To alleviate occasional overfitting, an early stop is applied when the model achieves a convergence under a threshold of 0.05. In our experiments, we ran the training set for a few iterations to record the approximate ratio z of the frequentist loss and Bayesian loss. Then, the corresponding weights W_f and W_b as per the total loss (11) are set respectively to 1 and 1/z.

A. Datasets

In data preparation, some open-source datasets from other state-of-the-art papers and aerial photography using UAVs are utilized for the experiments. The developed UAV system [25] forms the Skynet for inspection and imaging, under the consistent communication to the ground base. Those UAVs cruise around the surface of an infrastructure asset and take photos using a GoPro Hero 6 camera mounted on the drone.

The dataset used for training and validation is the CFD [26].

- *CFD* [26]: The dataset with 118 images of road cracks with labeled masks is used for training and validation. To increase the number of samples, images are rotated with a range from 0 to 90 degrees, flipped vertically and horizontally, and randomly cropped to a size of 256×256. All the augmented data is then split into the training and the validation set with a ratio of 9:1.

To verify the detection quality of the compared models, we form the testing set from four datasets unrelated to the training set:

- *Crack500* [27]: containing 500 images of pavement cracks with the benchmarks annotated by civil engineers. The images are resized into 256×256 to suit the limited computation power.
- *SYDCrack* [4]: An image dataset that was collected by our UAV system for detection of surface cracks, containing 850 image pitches with a size of 224×224.

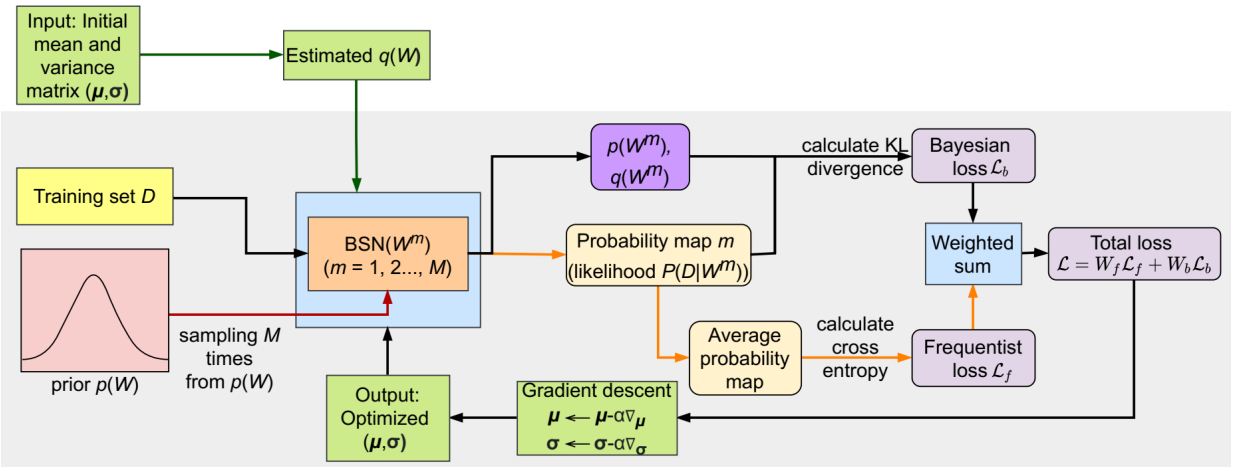

Fig. 2: Bayesian inference embedded in BSNBM training.

- *DCD* [28]: containing 521 images of infrastructure cracks with texture and misleading marks.
- *GAPs* [29]: containing 509 images of pavement cracks under poor light conditions. DCD and GAPs are both integrated into an unified size of 448×448 by [30].

To verify effectiveness of the proposed bidirectional self-rectifying network with Bayesian modeling (BSNBM), a comparison upon various databases of civil structure cracks is conducted between the existing crack detection approaches using frequentist deep learning and our BSNBM approach.

The frameworks for crack detection used in the comparison are described in the following:

- *DeepCrack* [2]: An end-to-end network for crack extraction using the backbone of SegNet with symmetrical concatenation.
- *FPHBN* [27]: It is a hierarchical framework for crack detection and constructed on the main framework of the holistically nested edge detection (HED) [31], enhanced using a dense connection.
- *CrackNetV* [32]: A sequential network for crack detection with a specific activation function for crack detection. No downsampling process is involved inside this architecture.
- *FCN* [33]: A simplified encoder-decoder shaped in an hourglass structure for crack detection. It is backboned by SegNet with only 3 blocks on each side.
- *U-Net* [34]: A modified auto-encoder with a U-shape merging pipeline. In U-Net, the outputs of each encoder are concatenated with corresponding outputs of the decoder. The concatenated tensor is then further processed by the next decoder block.
- *PGA-Net* [35]: A network built on a pre-trained encoder with a similar pyramid feature fusion as FPHBN.
- *HDCB-Net* [36]: A modified U-Net with serial dilation.
- *HCNNFP* [4]: A modified hierarchical network built upon the backbone of DeepCrack with feature preserving modules.
- *HACNet* [37]: An unidirectional hierarchical network built on modified dense blocks from [38].
- *ACS* [39]: A U-Net crack detector trained by an adaptive cost-sensitive loss function.

For an ablation study of the proposed Bayesian model, our BSN without Bayesian modeling is included in the comparison.

B. Evaluation Metrics

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

F-measure (F_β): The probability map is evenly binarized by a threshold 0.5, where pixels with a probability greater (smaller) than 0.5 are categorized as imperfect or intact regions. For a single image, F-measure F_β [40], a generalized intersection to union metric [41], is used. The F_β is calculated as

$$F_\beta = (1+\beta^2)\frac{p \times r}{\beta^2 \times p + r}, \tag{25}$$

where p and r are respectively the precision and recall, based on the correctly-reported and falsely-reported positive or negative results; and β^2 denotes the weight between precision p and recall r. Here, β^2 are set as 0.3 or 1 as recommended in [27] and [42] respectively. The score treats p and r equally when $\beta^2 = 1$, but favors to the prediction with more false-positive cases when $\beta^2 = 0.3$. A larger F-measure F_β indicates higher performance of the classification.

Average F-measure (AF_β): For robustness in evaluation with the F-measure, we also use the average F-measure AF_β, defined over a given range of β^2 as [4]:

$$AF_\beta = \frac{1}{\beta_2^2 - \beta_1^2}\int_{\beta_1^2}^{\beta_2^2} F_\beta d\beta^2, \tag{26}$$

where β_1^2 and β_2^2 respectively indicate the lower and upper limit for the range of β^2. Generally, β^2 is nonzero while it should be smaller than 1 if we emphasize the precision. Accordingly, we choose $\beta^2 \in (0, 1)$ as the range of the integration. Similarly to F_β, the higher AF_β the better performance.

Jaccard Index (JI): JI [40] is a criterion to judge the fitness between the predicted result and the ground truth.

Mathematically, the Jaccard index is equal to the ratio between the number of true-positive samples and the sum of true-positive, false-positive and false-negative samples. Compared

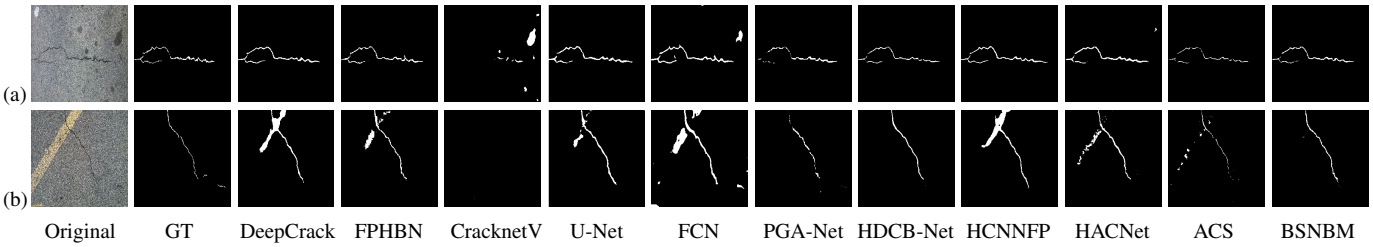

Fig. 3: Detection results on *Crack500* against granular surfaces.

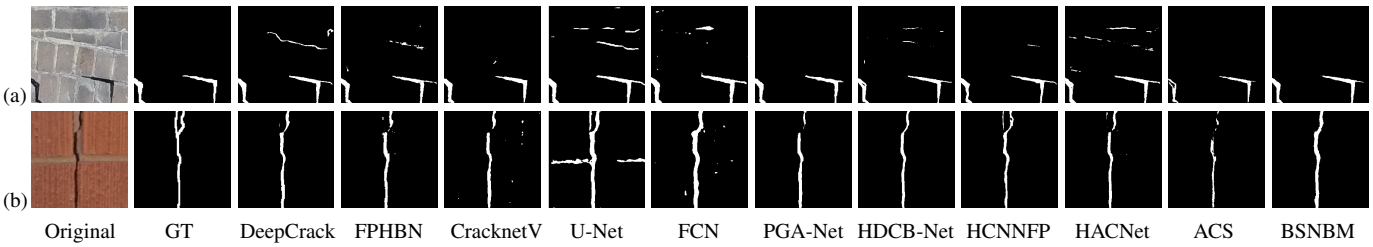

Fig. 4: Detection results on *SYDCrack* against regular texture.

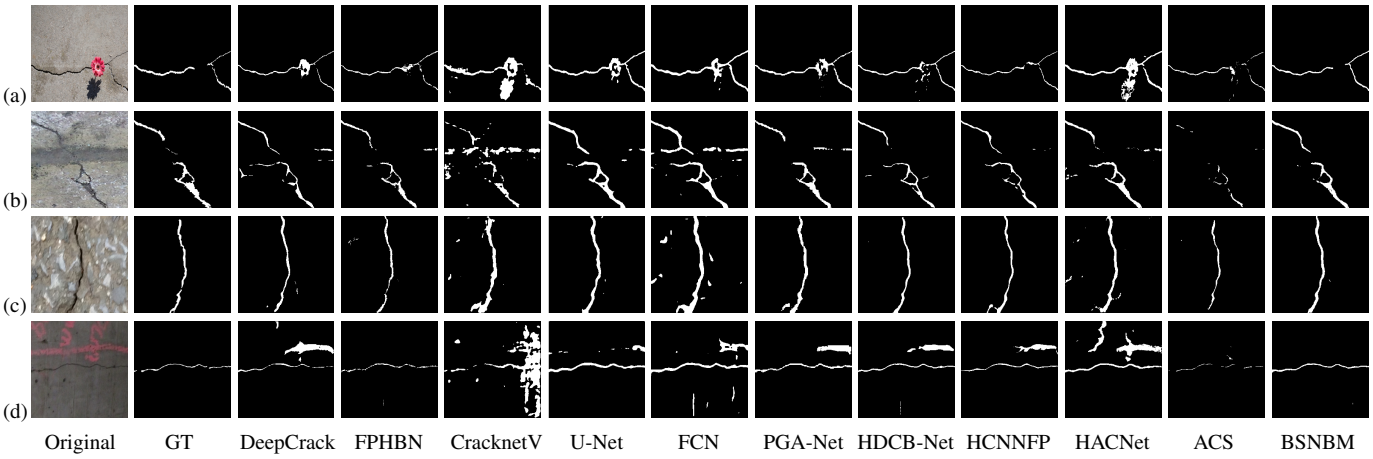

Fig. 5: Detection results on *DCD* against confusing marks and irregular texture.

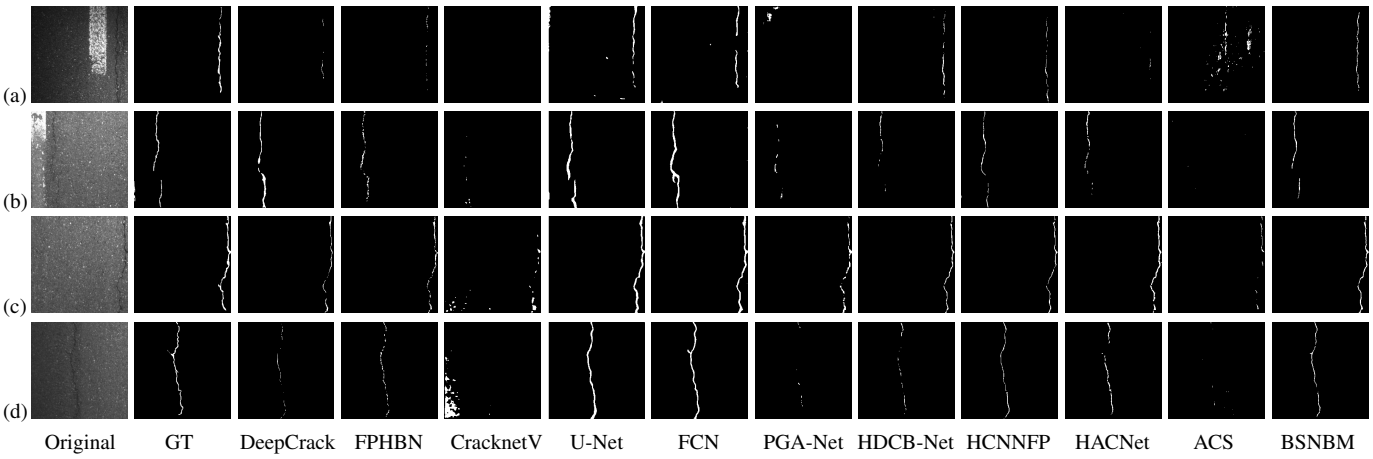

Fig. 6: Detection results on *GAPs* against poor light conditions

with F_β, the metric JI values more on balanced results with similar numbers of false-positive and false-negative predictions.

MAPE: For a binary map $S = \{s_{ij}\}$, the mean absolute percentage error (MAPE) is defined as,

$$MAPE = \frac{1}{N_{tp}} \sum_{i=1}^{I} \sum_{j=1}^{J} |s_{ij} - y_{ij}|, \tag{27}$$

where N_{tp} is the number of crack pixels in the ground truth (true-positive).

Compared with the mean absolute error, MAPE can differentiate the tested approaches better since the number of crack features from robotic photography is usually much smaller than the number of the background.

A smaller *MAPE* indicates a better match to the ground truth. Complementarily to the weighted F-measure, this metric rewards prediction with a high recall rate due to its preference for false-positive samples.

V. Results and Discussion

This section presents visual and quantitative comparisons of the tested approaches as well as critical discussion upon the experimental results.

A. Visual Comparison

The sample results of *Crack500* are presented in Fig. 3 for the compared approaches. It can be seen that without recursive connections like self-rectifying modules, it is more difficult to distinguish the crack-like patterns from real cracks using CrackNet-V. Consequentially, the noise in those sequential models has a severer effect than in other architectures. Relatively, hierarchical models like Deepcrack, FPHBN, U-net and the proposed BSNBM are more stable in dealing with those confusing scenes as most of the Frequentist models seem not affected by non-crack objects such as paints and shadows. The multi-level comparison between the feature maps and the ground truth contributes positively in terms of the accuracy of the prediction. Apart from that, the BSNBM presents a rather completed contour of the crack with clear dark dots. As shown in the second image, the unbroken line on the pavement strongly affects some models. The rationale for Bayesian inference is that our BSNBM can accurately remove the paint mark with good preservation of the crack silhouette.

Similarly, detection results of *SYDcrack* can also be observed consistently with *Crack500*. As exemplified in the first image, darker intensity pixels from dents or the texture displayed on images captured by robotic vehicles are quite challenging for compared models. It is not surprising that the level of noise in CrackNet-V and FCN still dominates other pipelines. As can be seen, the performance of hierarchical models is better in terms of resistance to the same noise. However, the effect of such interference is significantly mitigated in the proposed BSNBM. Even in clear scenes, the Bayesian model depicts a thinner contour of the crack with fewer adjacent false-positive samples classified as crack. Such difference in performance can be made in credit of Bayesian inference incorporated in Algorithm 1. Since Bayesian models are known for reliability, e.g. with respect to unfamiliar scenes in this case, it is less likely to recognize those confusing pixels as cracks unless more extensive samples are supporting this decision. As such, Bayesian models can achieve a higher recall rate compared with frequentist models.

B. Quantitative Comparison

The F-measure obtained from the mentioned approaches upon four datasets are listed in Table I. With different values of β, our bidirectional self-rectifying network with Bayesian modeling (BSNBM) outperforms other approaches in all the datasets in terms of F_β and AF_β. The proposed BSNBM performs as the best followed by another current frequentist model, the HDCB-Net. Compared with HDCB-Net, Bayesian inference displays some improvement on $F_\beta|_{\beta^2=1}$ especially in *GAPs* by 4.66%. The $F_\beta|_{\beta^2=0.3}$ on GAPs are an exception but it is just slightly lower than HDCB-Net with a gap of 0.15%. It is worth noting that $F_\beta|_{\beta^2=1}$ values more on precision compared with $F_\beta|_{\beta^2=0.3}$ so the results of it prove the capacity of our model in terms of reducing false-positive predictions. This indicates the effectiveness of the proposed model over existing approaches in distinguishing crack-like objects.

The performance criteria JI and $MAPE$ of the samples used for testing from four datasets are depicted respectively in Fig. 3 to Fig. 6. Obviously, the proposed BSNBM achieves the highest JI and lowest $MAPE$ with regards to almost all those samples.

Furthermore, as tabulated in Table II, the results of JI and $MAPE$ for all eleven approaches in consideration suggest that our Bayesian model also achieves the best performance in terms of the overall statistics. In terms of F_β, the recall rate contributes more positively to JI and $MAPE$. Accordingly, the quantitative results of JI and $MAPE$ indicate the improvement of the Bayesian model in minimizing false-negative samples. In other words, the Bayesian model is more capable of picking crack candidates out of a confusing background such as adjacent areas around the crack contour. Combining those criteria with F_β and AF_β, the quantitative results confirm the effectiveness of the proposed BSNBM. It can also be noted that the merit from the Bayesian model is more obvious in challenging datasets like *GAPs*. Such adaptive improvement contributes to the smallest performance gap of the Bayesian model between different datasets among other pipelines in the comparison, which points to a prominent advantage of the Bayesian one, i.e., robustness. Additionally, the floating point operations (FLOPs) and the average processing time (APT) of the above models spent on a 256×256 test sample are listed in Table III for the comparison of memory consumption and processing time, respectively. Among the top three approaches, our FLOPs consumption is only about 60% of HDCB-Net and 50% of HCNNFP and just behind the recently-developed HACNet. However, the proposed BSNBM is better than the HACNet in terms of APT. These indicate the computational efficiency of our proposed approach.

C. Failure cases

Extensive experiments may lead to some failure cases. As shown in Fig. 7, our proposed model is misled by the shadowy boundary lines wherein the dark pixels are incorrectly annotated as cracks. Unlike the successful cases presented in Fig. 5, the low contrast in intensity or color between marginal pixels here could have an impact on the prediction. Since

Methods	*Crack500*			*SYDCrack*			*DCD*			*GAPs*		
	β^2=0.3 ↑	β^2=1 ↑	AF_β ↑	β^2=0.3 ↑	β^2=1 ↑	AF_β ↑	β^2=0.3 ↑	β^2=1 ↑	AF_β ↑	β^2=0.3 ↑	β^2=1 ↑	AF_β ↑
DeepCrack [2]	80.83%	82.19%	81.22%	85.14%	85.39%	85.22%	86.56%	85.68%	86.32%	76.57%	72.19%	75.38%
FPHBN [27]	82.07%	81.85%	82.02%	85.17%	86.01%	85.42%	86.53%	85.15%	86.15%	78.16%	72.52%	76.62%
CrackNetV [32]	74.84%	68.75%	73.20%	81.31%	84.24%	82.15%	79.22%	82.37%	80.13%	58.51%	58.36%	58.48%
U-Net [34]	75.42%	78.84%	76.29%	77.78%	82.30%	79.07%	78.97%	83.49%	80.26%	68.31%	71.65%	69.27%
FCN [33]	80.42%	81.60%	80.76%	82.74%	85.68%	83.58%	80.49%	84.70%	81.70%	76.19%	78.48%	76.85%
PGA-Net [35]	82.31%	80.96%	81.94%	85.06%	86.11%	85.37%	86.10%	87.77%	86.58%	76.01%	70.81%	74.60%
HDCB-Net [36]	82.97%	80.69%	82.34%	85.80%	85.62%	85.77%	87.68%	87.91%	87.75%	**81.98%**	75.34%	80.17%
HCNNFP [4]	81.82%	82.05%	81.88%	85.51%	85.70%	85.58%	86.88%	85.94%	86.62%	79.33%	74.75%	78.07%
HACNet [37]	82.01%	81.51%	81.88%	85.56%	85.27%	85.50%	85.55%	88.23%	86.32%	80.97%	77.35%	79.98%
ACS [39]	77.27%	71.16%	75.61%	84.77%	78.64%	83.11%	86.62%	80.78%	85.02%	59.43%	55.69%	58.44%
BSNBM	**83.07%**	**82.72%**	**82.98%**	**85.92%**	**86.52%**	**86.10%**	**87.71%**	**88.79%**	**88.02%**	81.83%	**79.70%**	**81.25%**

TABLE I: Comparison of AF_β obtained from all nine detection networks.

Methods	*Crack500*		*SYDCrack*		*DCD*		*GAPs*	
	JI ↑	$MAPE$ ↓	JI ↑	$MAPE$ ↓	JI ↑	$MAPE$ ↓	JI ↑	$MAPE$ ↓
DeepCrack [2]	49.75%	89.46%	53.26%	69.94%	56.87%	76.49%	25.06%	96.17%
FPHBN [27]	49.05%	77.82%	55.23%	68.63%	55.26%	76.04%	24.43%	87.23%
CrackNetV [32]	27.04%	98.91%	50.56%	97.76%	48.95%	207.12%	8.84%	379.49%
U-Net [34]	40.20%	137.83%	44.66%	129.26%	47.98%	160.89%	25.76%	186.14%
FCN [33]	48.31%	92.51%	54.16%	85.00%	51.49%	149.50%	38.88%	118.41%
PGA-Net [35]	46.89%	73.03%	55.23%	70.25%	61.74%	78.02%	21.98%	91.80%
HDCB-Net [36]	47.60%	79.95%	53.96%	66.31%	61.61%	72.07%	28.55%	78.14%
HCNNFP [4]	49.51%	80.81%	54.30%	67.25%	57.08%	75.20%	29.68%	85.03%
HACNet [37]	50.72%	73.78%	52.89%	68.63%	62.08%	97.60%	35.37%	78.61%
ACS [39]	23.03%	88.57%	35.57%	71.42%	41.83%	65.77%	22.00%	120.28%
BSNBM	**51.20%**	**71.95%**	**56.31%**	**65.57%**	**64.15%**	**67.99%**	**41.29%**	**75.73%**

TABLE II: Comparison of JI and $MAPE$ among nine crack detection networks.

Methods	DeepCrack	FPHBN	CrackNetV	U-Net	FCN	PGA-Net	HDCB-Net	HCNNFP	HACNet	ACS	BSNBM
FLOPs(G)	42.54	22.28	14.49	42.83	24.41	53.68	37.36	43.51	21.70	42.83	21.98
APT(ms)	29.67	20.28	8.81	35.14	20.84	103.26	36.38	30.09	25.14	35.14	21.95

TABLE III: Comparison of FLOPs and APT among nine DCNN approaches.

the intensity and curvature of those marginal pixel groups are quite similar to the majority of truth-positive samples obtained from the training set, the expectation of weights collected in Bayesian inference may remain close to the original frequentist values and become incapable of bringing any significant improvement.

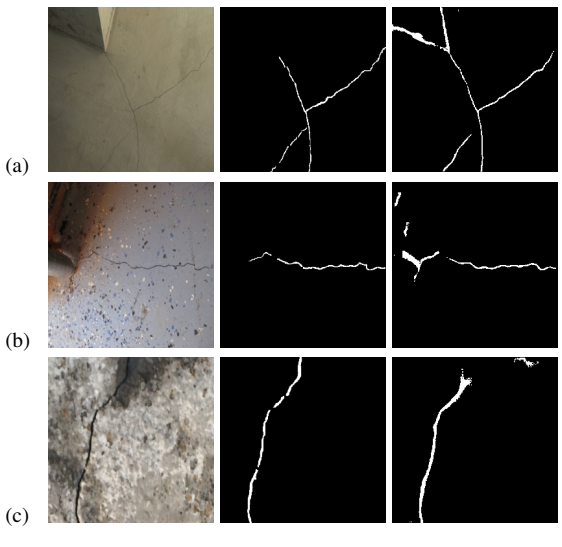

Fig. 7: Failure cases

A comparison is further conducted between the prediction results on the original image shown in Fig. 7 (c) and its preprocessed image with adjusted hue or brightness. As demonstrated in Fig. 8, the missing patterns are retrieved after the hue of the image is switched along a specific direction while overall intensity is unchanged. This means that the color contrast can directly affect the prediction result. Since the training data is not even-colored or grayscaled, it is inevitable that the model finally forms its preference on a specific color channel.

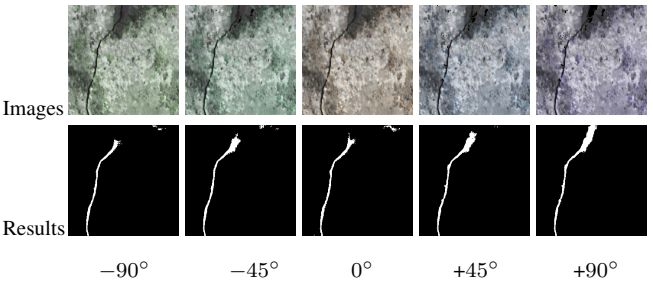

Fig. 8: Detection results on different hue degrees (0°: Original hue)

The detection results of the proposed model on different brightness levels are presented in Fig 9. Therein, the prediction accuracy is increasing with the rise of brightness, which indicates the potential improvement with light adjustment. One possible reason is that the training set CFD [26] is collected under a good light condition. As such, the trained model appears to be more sensitive to a higher average intensity.

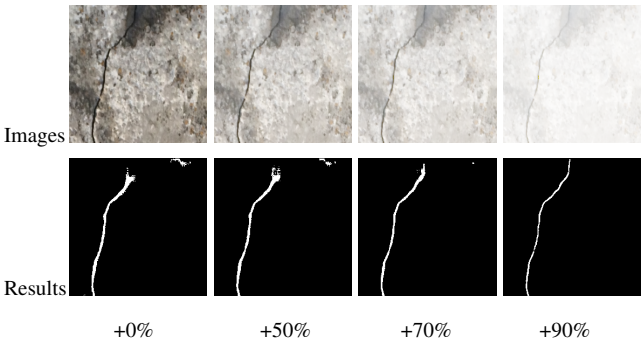

Fig. 9: Detection results on different brightness levels (+0%: Original brightness)

D. Ablation Study

A comprehensive ablation study of frequentist network elements (bidirectional self-rectifying modules, number of dilated convolutional blocks, and loss functions) as well as inference types (with or without Bayesian model) is conducted on the GAPs dataset in various cases for all the compared approaches.

1) Ablation on Bidirectional Self-Rectifying Modules: For bidirectional self-rectifying modules, cases are compared upon several k scales of abstractions, namely unilaterally (U) and bidirectionally (B), or without applying self-rectifying modules (Nan). As shown in Table IV, with the help of these modules, the AF_β and JI are improved by 2.78% and 8.68% respectively while $MAPE$ drops by 7.96%, indicating the contribution of the proposed bidirectional rectification. Here, thanks to more contextual information being processed by the bidirectional branches, the abstractions containing inconsistent crack patterns like dots are more likely to possess a lower ratio/weight in overall representation. Therefore, such a pattern can trigger a smaller response in the gradient update, which is beneficial to address overfitted outliers.

SRM	$F\|_{\beta^2=0.3}\uparrow$	$F\|_{\beta^2=1}\uparrow$	$AF_\beta\uparrow$	$JI\uparrow$	$MAPE\downarrow$
Nan	79.33%	74.75%	77.12%	29.68%	85.03%
U	78.64%	77.47%	77.94%	35.19%	78.55%
B	**80.22%**	**78.32%**	**79.70%**	**38.36%**	**77.07%**

TABLE IV: Ablative comparison on self-rectifying modules

2) Ablation on Number of DCBs: To investigate the influence of the DCB quantity, our standard 3-DCB BSN is compared with other numbers of dilated convolutional blocks. As shown in Table V, the current setup with 3 DCBs outperforms other configurations. With one DCB removed, the AF_β and JI decrease by 4.47% and 5.61% respectively while $MAPE$ increases by 8.04%. This is because the total number of kernels drops dramatically with a smaller number of DCBs. In contrast, the 4-DCB network just improves JI slightly by 1.42%, while the AF_β declines by 2.03% and $MAPE$ increases by 16.3% unexpectedly, indicating ineffectiveness, perhaps from a more complex gradient space for parameter optimization. Despite enhancing the capacity in feature preservation, such complexity negatively influences the convergence to the benchmark and tends to degrade the network performance. One possible reason could be that the increasing hierarchy of abstraction on one hand enhances the comprehensiveness of benchmarking but on the other hand poses some difficulty in the gradient descent due to the accumulating complexity of abstracted tensors. When the number of DCBs is growing, the deviation from the optimization process may be dominant, superseding the benefit of the enhanced branches, and thus, leading to a drop in quantitative performance.

DCBs	$F\|_{\beta^2=0.3}\uparrow$	$F\|_{\beta^2=1}\uparrow$	$AF_\beta\uparrow$	$JI\uparrow$	$MAPE\downarrow$
2	74.43%	73.68%	74.23%	30.48%	133.88%
3	**80.22%**	**78.32%**	**79.70%**	38.36%	**77.07%**
4	77.55%	77.65%	77.67%	**39.77%**	93.37%

TABLE V: Ablative comparison on DCBs

3) Ablation on Frequentist Loss Functions: In regard to loss functions, we compare the binary cross entropy (BCE) with alternative criteria such as mean absolute error (MAE) and mean square error (MSE) [43]. As shown in Table VI, the BCE loss shows its merit over other loss functions. Using the BCE, the AF_β increases by 2.46% and 3.30% respectively with respect to MAE- or MSE-based learning scenario, while JI and $MAPE$ grow even more significantly. This indicates that BCE is the best choice for the frequentist loss function as per the criteria used. Its rationale could be that the gradient vanishes for the BCE loss with the softmax function, a general case of sigmoid functions, and thus leading to a slowdown or stagnation in the gradient update [44].

Loss	$F\|_{\beta^2=0.3}\uparrow$	$F\|_{\beta^2=1}\uparrow$	$AF_\beta\uparrow$	$JI\uparrow$	$MAPE\downarrow$
MAE	78.36%	74.25%	77.24%	28.90%	98.92%
MSE	76.83%	75.21%	76.40%	32.22%	128.38%
BCE	**80.22%**	**78.32%**	**79.70%**	**38.36%**	**77.07%**

TABLE VI: Ablative comparison on loss functions

4) Ablation on Types of Inference: For ablation between frequentist and Bayesian inference, the proposed BSNBM is compared with its frequentist network without Bayesian modeling (BSN). As shown in Table VII, thanks to Bayesian modeling, the AF_β and JI increase by 1.55% and 3.13% respectively while $MAPE$ is reduced by 1.34%. This clearly indicates the merit of Bayesian modeling for crack detection applications. This can be explained by the capability of Bayesian models in dealing naturally with uncertainty estimation via the probabilistic description.

Methods	$F\|_{\beta^2=0.3}\uparrow$	$F\|_{\beta^2=1}\uparrow$	$AF_\beta\uparrow$	$JI\uparrow$	$MAPE\downarrow$
BSN	80.22%	78.32%	79.70%	38.36%	77.07%
BRNBM	**81.83%**	**79.70%**	**81.25%**	**41.29%**	**75.73%**

TABLE VII: Ablative comparison on inference types

VI. Conclusion

This paper has presented a novel deep learning approach for crack detection. The proposed hierarchical neural network is enhanced by residual modules, which improves the network's capability of feature preservation. To further achieve higher accuracy and robustness, the network is integrated with Bayesian inference. The Bayesian model is developed

incorporating extensive evidence for the true-positive rate of crack detection under unfamiliar scenes. Hence, the model can better identify a true crack with a comprehensive decision in the alleviation of overfitting, which is commonly encountered in frequentist networks. Consequently, the proposed approach can successfully perform robotic crack detection in civil infrastructure applications as demonstrated by using four heterologous datasets with different levels of textures and light conditions. Results from various tests and a thorough comparison with current frequentist deep learning pipelines for crack detection have been included to indicate the merits of the proposed framework. Its extension of the performance comparison for other probabilistic distributions besides the Gaussian one will be our future work.

REFERENCES

[1] H. Kim, E. Ahn, S. Cho, M. Shin, and S.-H. Sim, "Comparative analysis of image binarization methods for crack identification in concrete structures," *Cement and Concrete Research*, vol. 99, pp. 53–61, 2017.

[2] 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.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. of the IEEE conf. computer vision and pattern recognition*, 2016, pp. 770–778.

[4] Q. Zhu, T. H. Dinh, M. D. Phung, and Q. P. Ha, "Hierarchical convolutional neural network with feature preservation and autotuned thresholding for crack detection," *IEEE Access*, vol. 9, pp. 60 201–60 214, 2021.

[5] C. J. Spoerer, P. McClure, and N. Kriegeskorte, "Recurrent convolutional neural networks: A better model of biological object recognition," *Frontiers in Psychology*, vol. 8, p. 1551, 2017.

[6] M. Attia, M. Hossny, S. Nahavandi, and H. Asadi, "Surgical tool segmentation using a hybrid deep cnn-rnn auto encoder-decoder," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2017, pp. 3373–3378.

[7] X. Yang, H. Mei, J. Zhang, K. Xu, B. Yin, Q. Zhang, and X. Wei, "Drfn: Deep recurrent fusion network for single-image super-resolution with large factors," *IEEE Transactions on Multimedia*, vol. 21, no. 2, pp. 328–337, 2018.

[8] W. Luo, Y. Li, R. Urtasun, and R. Zemel, "Understanding the effective receptive field in deep convolutional neural networks," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 2016, pp. 4905–4913.

[9] 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.

[10] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[11] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[12] 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.

[13] Y. Wei, H. Xiao, H. Shi, Z. Jie, J. Feng, and T. S. Huang, "Revisiting dilated convolution: A simple approach for weakly-and semi-supervised semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7268–7277.

[14] S. Ghosh, A. Dhall, and N. Sebe, "Automatic group affect analysis in images via visual attribute and feature networks," in *2018 25th IEEE Int. Con. on Image Processing (ICIP)*. IEEE, 2018, pp. 1967–1971.

[15] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*, 2016, pp. 1050–1059.

[16] T. Chen, R. Xu, Y. He, and X. Wang, "Improving sentiment analysis via sentence type classification using bilstm-crf and cnn," *Expert Systems with Applications*, vol. 72, pp. 221–230, 2017.

[17] Y. Wang, F. Liu, K. Zhang, G. Hou, Z. Sun, and T. Tan, "Lfnet: A novel bidirectional recurrent convolutional neural network for light-field image super-resolution," *IEEE Transactions on Image Processing*, vol. 27, no. 9, pp. 4274–4286, 2018.

[18] L. Mou, P. Ghamisi, and X. X. Zhu, "Deep recurrent neural networks for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 7, pp. 3639–3655, 2017.

[19] W. K. Hastings, "Monte carlo sampling methods using markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.

[20] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American statistical Association*, vol. 112, no. 518, pp. 859–877, 2017.

[21] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural network," in *International Conference on Machine Learning*, 2015, pp. 1613–1622.

[22] K. Shridhar, F. Laumann, and M. Liwicki, "A comprehensive guide to bayesian convolutional neural network with variational inference," *arXiv preprint arXiv:1901.02731*, 2019.

[23] Y. Wu and K. He, "Group normalization," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[24] 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, pp. 1–15.

[25] 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*, vol. 14, no. 2, pp. 2925–2936, 2020.

[26] 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.

[27] 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, 2019.

[28] 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.

[29] 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 *International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 2039–2047.

[30] 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.

[31] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proceedings of the IEEE Inter. Conf. Computer Vision*, 2015, pp. 1395–1403.

[32] 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*, vol. 21, no. 1, pp. 273–284, 2019.

[33] 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.

[34] 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.

[35] 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*, vol. 16, no. 12, pp. 7448–7458, 2019.

[36] W. Jiang, M. Liu, Y. Peng, L. Wu, and Y. Wang, "Hdcb-net: A neural network with the hybrid dilated convolution for pixel-level crack detection on concrete bridges," *IEEE Trans. Industrial Informatics*, 2020.

[37] H. Chen and H. Lin, "An effective hybrid atrous convolutional network for pixel-level crack detection," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–12, 2021.

[38] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[39] K. Li, B. Wang, Y. Tian, and Z. Qi, "Fast and accurate road crack detection based on adaptive cost-sensitive loss function," *IEEE Transactions on Cybernetics*, 2021.

[40] H.-H. Chang, A. H. Zhuang, D. J. Valentino, and W.-C. Chu, "Performance measure characterization for evaluating neuroimage segmentation algorithms," *Neuroimage*, vol. 47, no. 1, pp. 122–135, 2009.

[41] 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/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 658–666.
[42] J.-J. Liu, Q. Hou, and M.-M. Cheng, "Dynamic feature integration for simultaneous detection of salient object, edge, and skeleton," *IEEE Transactions on Image Processing*, vol. 29, pp. 8652–8667, 2020.
[43] J. O. Berger, *Statistical decision theory and Bayesian analysis*. Springer Science & Business Media, 2013.
[44] P. Golik, P. Doetsch, and H. Ney, "Cross-entropy vs. squared error training: a theoretical and experimental comparison." in *Interspeech*, vol. 13, 2013, pp. 1756–1760.

Qiuchen Zhu (S'18) 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.

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

He is currently an Associate Professor with the Faculty of Engineering and Information Technology, University of Technology Sydney, Australia. His research interests include automation, robotics, and control systems. He received several paper awards from the IEEE, IAARC, and Engineers Australia, including the 2015 Sir George Julius Medal. He chaired 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), Mathematical Problems in Engineering, and Electronics. He is currently an Associate Editor of Automation in Construction and Robotica.