BIDIRECTIONAL SELF-RECTIFYING NETWORKS WITH BAYESIAN MODELLING FOR FEATURE DETECTION AND KEYPOINT ALLOCATION

QIUCHEN ZHU[1], **QUANG HA**[1]

[1]School of Electrical and Data Engineering, Faculty of Engineering and IT, University of Technology Sydney, Australia
E-MAIL: Qiuchen.Zhu@student.uts.edu.au, Quang Ha@uts.edu.au

Abstract:

In machine vision, deep learning frameworks are getting more attractive to researchers owing to their accuracy and robustness for feature extraction. However, the uncertainty in data or model has an adversary impact on the prediction and limits the performance of deep learning. To address the problem associated with uncertainty, we propose a bidirectional self-rectifying network with Bayesian modelling (BSNBM) for feature detection. First, a set of branch networks is proposed, wherein the output of previous convolutional blocks is unified and concatenated to the current ones to reduce the visual impairment in the up/down-sampling stage, taking into account the overall information loss. Further, our framework is probabilistically based on Bayesian modelling using prior knowledge. In the Bayesian model, the weight of the learnable layers are converted into distribution functions. Such conversion aims to improve robustness against outliers and therefore alleviate the overfitting issue. The proposed technique is then applied to identify surface cracks of infrastructure such as roads, bridges or pavements. Extensive comparison with existing techniques is conducted on various datasets, subject to a number of evaluation criteria. Experiments on crack images, including those captured by unmanned aerial vehicles inspecting a monorail bridge, demonstrate the merits of the proposed BSNBM architecture over existing techniques for surface defect inspection. Additional tests on extensive applications show the scalability and robustness of this model for various image processing tasks.

Keywords:

Deep learning; Bayesian inference; hierarchical convolutional neural network; image feature detection

1. Introduction

In computer vision, the features of an image is a fundamental element of semantic representation. When a group of pixels of an image forms a specific geometrical feature, the contrast and geometrical correlation of those pixels provide abundant information for reasoning in various pattern recognition. For typical image processing tasks, e.g. image classification and keypoint allocation, detection accuracy often rests with quality of the extracted feature. For those tasks, the first and most important step is the detection of semantic features. Then, further induction can be conducted by using the detected features as a visual clue.

Due to the variety of homogeneous patterns and ambiguity of semantic features, especially in dealing with abstract features required for saliency detection, the task of feature detection could be challenging. Indeed, the shape of salient features varies dramatically with the targets, making it difficult to find the graphical similarity of the features. Therefore, it requires a highly robust algorithm to extract the feature patterns. In this regard, with advances in neural computing, deep convolutional neural networks (DCNN) have emerged as a statistic framework that can effectively address this requirement and ideally cater for improvements in robustness and accuracy by learning from data.

In DCNN, overfitting is a common problem that limits detection performance in terms of robustness. In practice, the reliability of feature detection can be improved via a better exploitation on the given data. Therefore, researchers have devoted remarkable effort in improving fitness of the prediction. Despite a number of frameworks available, there are still some issues to be solved. First, the accuracy of detected features is affected by the unknown information that could be effectively utilised by DCNN. Secondly, the credibility of the training data actually changes with sampling; e.g., the benchmark features under a low resolution should be less trustable to the data processor but are treated equally as all the training samples in DCNN. Finally, mislabelled annotation of benchmark data could happen due to system's errors introduced from the data collection.

For a reliable extraction of features in DCNN, it is required

to address these issues, considering known as well as unknown information. Only a few techniques have been proposed to tackle what is unknown in DCNN. Nonetheless, such veiled information is an important factor affecting the quality of feature detection and could be used to potentially improve DCNN performance. For that, this paper aims to explore, using the statistic approach, about uncertainty in DCNN. To this end, the structure of DCNN for feature extraction is reconsidered from a Bayesian modelling prospective in this paper. Here, a solution to handle uncertainty exploiting the hidden prior knowledge is proposed for robustness enhancement in feature detection.

2. Bidirectional self-rectifying networks for feature detection

2.1 Network architecture

The proposed bidirectional self-rectifying network (BSN) is shown in Fig. 1. Unlike standard hourglass-shape models [1] with 5 convolutional blocks per side, the main network here consists of only 3 dilated convolutional blocks (DCBs) instead with one dilated convolutional layer [2] sandwiched by two standard convolutional layers per block. Those blocks perverse the hierarchical abstractions of features in three different scales to be described in Section 2.3.

With the combination of several convolutional layers, the feature map is refined after each block. The convolutional abstraction from each DCB are further processed by a forward and a reverse enhancement branch (FEB/REB) bidirectionally and finally fed into a feature merging net to produce the final probability map of features by multi-scale fusion.

2.2. Logistic regression and probabilistic maps

The proposed network infers the logistic regression on an arbitrary training sample denoted as $\{(X,Y)\} = \{x_{ij}, y_{ij}|i,j \in (I \times J)\}$, where x_{ij} and y_{ij} respectively represent the pixel values of the original image and its corresponding annotated mask both in a size of $I \times J$. Accordingly, the ground-truth mask y_{ij} takes a binary value determined as,

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

The probability of an arbitrary pixel belonging to a feature $\{F|f_{ij}\}$ can be calculated by using the sigmoidal function,

$$P(f_{ij}) = \frac{1}{1+e^{-f_{ij}}}. \quad (2)$$

Theoretically, the sigmoid layer in CNN can represent the accurate probability map for arbitrary data when the feature f_{ij} transformed from the input x_{ij} is unbiased. However, as discussed in [3], such prerequisite can be hardly met in CNN due to nonlinearities in f_{ij}.

2.3 Bidirectional self-rectifying abstractions

To implement bidirectional abstractions of the feature by obtaining directional observations from different views, the network can generate a more balanced feature map based on multi-scale patterns. To take into account the difference in scales, the concatenation here is conducted with a resized operation. As shown in the forward branch of Fig. 1, the width of the larger tensor shrinks to a half size using an additional downsampling process. Two size-halved tensors are then merged and fed to the next layers with superimposed channels, which is contrary in the reverse branch.

With the merged feature maps, the associated information loss for feature $\{F^k|f_{ij}^k\}$ at the k^{th} convolutional block can be expressed via its entropy as,

$$l(f_{ij}^k) = -y_{ij}\ln(P(f_{ij}^k)) - (1-y_{ij})\ln(1-P(f_{ij}^k)). \quad (3)$$

Specifically in the proposed network, three feature maps each at a different scale of abstractions and an additional fused map are included in the loss function as,

$$\mathcal{L}_f = \sum_{i=1}^{I}\sum_{j=1}^{J}\left(l(f_{ij}^{fused}) + \sum_{k=1}^{3} l(f_{ij}^k)\right). \quad (4)$$

2.4. Bayesian inference

Let us consider the weights W of the convolutional kernels 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 as per the Bayes' theorem for a given training set $\mathcal{D}$:

$$p(W \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid W)\, p(W)}{\int p(\mathcal{D} \mid W)\, p(W) dW}. \quad (5)$$

For a new sample $\{x^*, y^*\}$, an unbiased estimation of the probability can be given by using Monte Carlo sampling [4] on the output of the network with M samples:

$$\mathbb{E}_{p(W|\mathcal{D})} p(y^* \mid x^*, W) \simeq \frac{1}{M}\sum_{m=1}^{M} p\left(y^* \mid x^*, W^{(m)}\right), \quad (6)$$

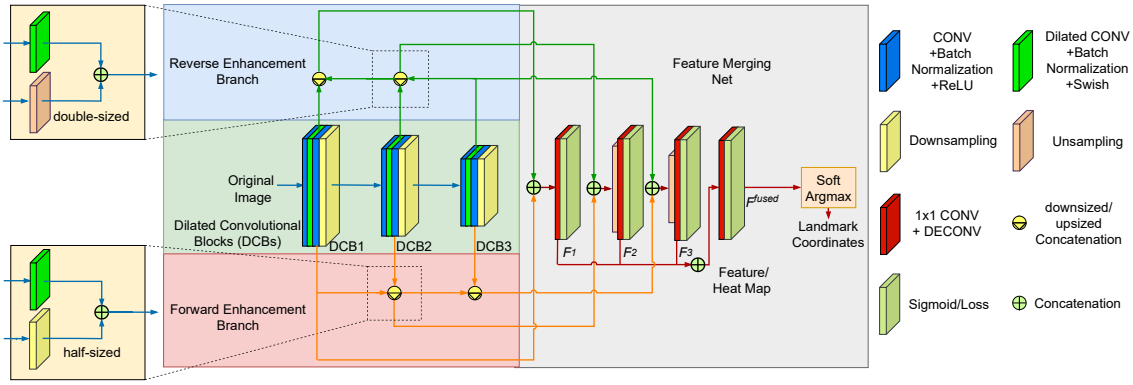

FIGURE 1. Architecture of the proposed network

where $W^{(m)}$ is randomly picked from the conditional distribution of W. Therefore, the posterior probability $p\,(W \mid X, Y)$ 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 a random Gaussian distribution $q_\phi(W)$, i.e. $p(W \mid \mathcal{D}) \approx q_\phi(W)$. The obtained parameters can be obtained using the same routine of the DCNN backpropagation.

2.5. Spatial indexing on feature/heat map

To take into account uncertainty in the allocation, the coordinates of landmarks are represented as a heat map by using a 2D-Gaussian kernel with a variance σ_g,

$$G(i,j) = \frac{1}{\sqrt{2\pi}\sigma_g} e^{-\frac{(\Delta i)^2+(\Delta j)^2}{2\sigma_g^2}}, \tag{7}$$

where Δi and Δj are the horizontal and vertical distance from an arbitrary point to the target.

The landmark allocation can be achieved via the argmax function in a 2D space. However, the non-differentiability of this function is a severe problem to the deep learning frameworks. Without a certain derivative, the parameters of functional layers cannot be updated via the descent gradient. So here we use an alternative, the spatial soft-argmax function [5] to estimate the index of the landmark within an acceptable range of accuracy. The core calculation of the spatial softmax function can be expressed as follows,

$$h_s(x_{ij}) = \frac{e^{\frac{x_{ij}}{c_t}}}{\sum_{k=1}^{I}\sum_{l=1}^{J} e^{\frac{x_{kl}}{c_t}}}, \tag{8}$$

where x_{ij} indicates the pixel value specifically at the position (i, j) on the heat map, I and J represent the width and height of the image, and c_t is a temperature constant to control the density of the distribution.

The soft-argmax function projected on the two axes is the expectations of the softmax function over x_{ij}. The center (i, j) can be estimated respectively as,

$$\tilde{i} = h_{sa,i}(x_{ij}) = \sum_{i=1}^{I}\sum_{j=1}^{J} h_s(x_{ij})i, \tag{9}$$

$$\tilde{j} = h_{sa,j}(x_{ij}) = \sum_{i=1}^{I}\sum_{j=1}^{J} h_s(x_{ij})j. \tag{10}$$

Since the network is trained to output a similar distribution as of the ground truth, the generated heat map should possess common properties with the benchmark one. As a result, the Gaussian kernel with a large standard deviation σ_g can lead to the flat pixel distribution of the generated heat map, which contributes negatively to the coordinate estimation in terms of accuracy. Therefore, direct training on sparse Gaussian kernels often yields inadequate accuracy of the feature representation.

The effective size of the kernel is determined by deviation σ_g, whereby a smaller σ_g would lead to a denser kernel, generally beneficial to an accurate allocation. However, an overdense kernel would mean all the responses on the map are near to 0 and may cause an extremely imbalanced prediction affecting the representation in the trained model. In this case, the size of the kernel should be properly designed to achieve the required accuracy upon a bearable data balance.

2.6. Loss function for allocation

Landmark detection aims to predict the positions of M functional key points defined on the object of concern. Given an image I, the 2D-landmark locations from the annotation and the deduction of the soft-argmax function are defined as vectors $L = \{L_m : (L_m^{(1)}, L_m^{(2)})\}$ and $\widehat{L} = \{\widehat{L}_m : (\widehat{L}_m^{(1)}, \widehat{L}_m^{(2)})\}$,

respectively for $m = 1, \ldots, M$. For balancing all components in the information loss, the coordinates are normalised to the range (-1,1) by the normalisation function $R(\cdot)$. Accordingly, the loss function with respect to the coordinates can be expressed in terms of mean square error as,

$$\mathcal{L}_{co} = \frac{1}{M} \sum_{m=1}^{M} \left\| R\left(\widehat{L}_m\right) - R\left(L_m\right) \right\|_F . \tag{11}$$

where $\|\cdot\|_F$ is the Frobenius norm. Therefore, the average loss over the heat map now becomes,

$$\mathcal{L}_{hms} = \frac{1}{M} \sum_{m=1}^{M} \mathcal{L}_{f,m}, \tag{12}$$

where the total loss of the model obtained as the superposition of the coordinate loss and the heat map loss as,

$$\mathcal{L}_{total} = W_{co}\mathcal{L}_{co} + \mathcal{L}_{hms}, \tag{13}$$

in which W_{co} is the weight of $\mathcal{L}_{total}$ to balance the model's dependency on the fitness of the heat map and coordinates.

2.7. Dissipated training

For an alternative way to train the model with limited positive samples, dissipation training is proposed for Gaussian regression. To understand the concept, let us take an example of a bonfire. When it starts to burn in the wild, we can easily notice the shining flame and therefore know roughly the bonfire location. As the heat energy continue dissipating with time, the visible light around the heat point became less diffused. As a result, the flame becomes darker but more concentrated. Finally, the centre of the ignition can be detected. However, if a match is flaming, it is impossible to observe the flame from a distance at the beginning due to the limited energy of heat diffusion. Similarly, starting with a dense kernel may lead to an all-negative representation due to few positive samples in the benchmark as shown in Figure 2 (a). Inspired by the dissipation described in the example, we design a variable training scenario where the effective size of the Gaussian kernel is decreasing with the training epochs like the heat diffusing with time. In the beginning, the ground-truth heat map with a large kernel is applied to a rough regression. After each epoch, the width of the kernel is reset by adjusting the deviation σ_g to the half size of the previous one. As demonstrated in Figure 2 (b), the predicted heat map is gradually centralised in the landmark position, and also avoids ending up with an empty map at the local minima. This is because the logistical distance between the two heat maps generated in adjacent epochs is smaller than their distance to the all-zero map. From adjusting the kernel size, the model can effectively bypass the local-minima trap.

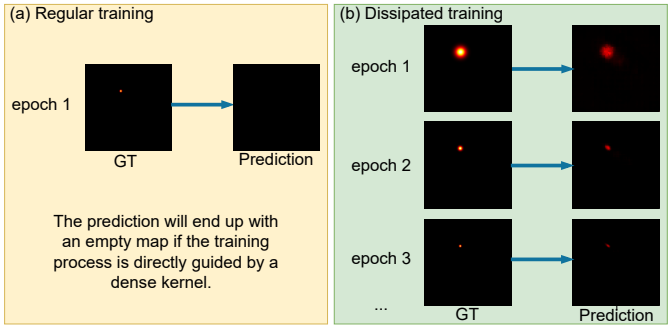

FIGURE 2. Heatmap training

3. Experimental results and discussion

To verify the effectiveness of the proposed approach, three experiments are conducted respectively on surface crack, salient object and landmark detection.

3.1. Datasets and algorithms in comparison

The public datasets for comparison are listed as follows:

- Crack detection: *DCD* [6], *GAPs* [7]. The experiment detail on those datasets can be referred in [3].
- Saliency detection: *MSRA10K* [8] and *MSRA-B* [8].
- Facial landmark detection: 300W [9].

The frameworks for the comparison of surface crack segmentation are listed in the following: *DeepCrack* [1], *FPHBN* [10], *PGA-Net* [11], *HDCB-Net* [12] and *HCNNFP* [3].

3.2. Evaluation metrics

3.2.1 Metrics for feature detection

In this paper, the following criteria from well recognised sources are used for evaluation.

- Average F-measure (AF_β) [3], Jaccard Index (JI) [13]: A larger value indicates a better segmentation result;
- The mean absolute percentage error (MAPE) [14]: A smaller *MAE* indicates a more accurate prediction.

3.2.2 Metrics for landmark detection

NME_{io}/NME_{ip}: The inter-ocular normalised mean error (NME_{io}) and the inter-pupil normalised mean error (NME_{ip})

[9] are specific metrics designed for the evaluation of facial landmark detection. A smaller value of the errors NME_{io} and NME_{ip} indicates a better match to the ground truth.

3.3. Results on Image Segmentation

3.3.1 Surface crack detection

The sample results are presented in Fig. 3 for the compared approaches. Relatively, hierarchical models like FPHBN, HDCB-Net and the proposed BSNBM performs more stably in dealing with those confusing scenes such as paints. 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 less false negative labels. The

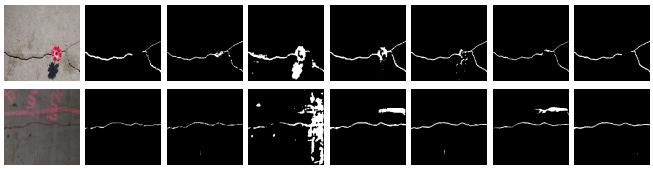

FIGURE 3. Comparison on Surface Defect Detection. From left to right: original image, ground truth, results respectively of FPHBN, PGA-Net, HDCB-Net, HCNNFP, and BSNBM.

Methods	*DCD*			*GAPs*		
	AF_β	JI	$MAPE$	AF_β	JI	$MAPE$
DeepCrack	86.32%	56.87%	76.49%	75.38%	25.06%	96.17%
FPHBN	86.15%	55.26%	76.04%	76.62%	24.43%	87.23%
PGA-Net	86.58%	61.74%	78.02%	74.60%	21.98%	91.80%
HDCB-Net	87.75%	61.61%	72.07%	80.17%	28.55%	78.14%
HCNNFP	86.62%	57.08%	75.20%	78.07%	29.68%	85.03%
BSNBM	**88.02%**	**64.15%**	**67.99%**	**81.25%**	**41.29%**	**75.73%**

TABLE 1. Comparison for crack detection results on *DCD* and *GAPs*.

quantitative results obtained from the compared approaches upon two datasets are shown in Table 1. Here AF_β is calculated with β^2 from 0 to 1. Our BSNBM outperforms over other approaches in all the datasets. HDCB-Net performs as the second best followed by another current frequentist model, HCNNFP.

3.3.2 Saliency Detection

The visual results of saliency detection are depicted in Fig. 4. Despite slight mislabelling in the boundary areas, the proposed approach is able to extract a highly matched saliency map, close to the instinctive attention. Compared with the ground truth, the silhouette of the predicted saliency map consists of simpler shapes as the proposed approach tends to use rather simple curves to link the detected endpoints on the image. The philosophy behind that is the BSNRM's robustness obtained from bidirectional abstractions and Bayesian inference. Such adjustment leads to a slight degradation in extracting marginal patterns due to mislabelled background pixels near the boundary of the salient object, and hence, is beneficial to robustness in deteting various salient objects.

FIGURE 4. Results on salient object detection. From left to right: original image, ground truth, and detection results respectively.

The quantitative results of the saliency detection are listed in Table. 2. On average, the AF_β and JI on MSRA10K and MSRA-B are respectively around 92% and 80%, which are quite high for saliency detection tests. These statistic metrics are also in accordance with the good performance of the visualised results as presented in Fig 4. The outperformance of the proposed approach applied to those primary tasks verifies its great potential for extensive low-level applications in crack or salient object detection.

Datasets	Metrics		
	AF_β	JI	$MAPE$
MSRA10K	92.88%	81.21%	24.94%
MSRA-B	91.86%	78.90%	41.62%

TABLE 2. Quantitative results for salient objective detection.

3.4. Results on Facial Keypoint Detection

The visual results of the proposed framework are depicted in Fig. 5, demonstrating the accuracy of the proposed bidirectional self-rectifying network with Bayesian modelling in landmark allocation as described above. The rationale behind it is the stochastic modelling in uncertainty can help the network overcome the influence of outlier samples. This contributes to robustness of the network against fluctuations in the data. Hence, under uncertainty conditions the allocation is more likely to yield correct samples from adopting the empirical knowledge.

Kernel Range	$\sigma_g = 1$	$\sigma_g = 2$	$\sigma_g = 4$
NME_{ip}	4.94%	5.98%	9.55%
NME_{io}	3.59%	4.35%	6.95%

TABLE 3. average error of facial landmark detection.

FIGURE 5. Visualized results of landmark detection using the proposed model. Red: predicted landmarks; Green: annotated landmarks

The quantitative results of facial landmark detection are presented in Table 3. As shown in this table, the proposed network achieves the most accurate prediction of landmarks when σ_g is set to be 1. The corresponding NME_{ip} and NME_{io} are 4.94% and 3.59% respectively, which are quite plausible in industrial applications. In practice, a simple alignment of designed modules can enable the proposed network to handle high-level tasks effectively, which is quite useful for industrial practitioners.

4. Conclusion

In this paper, we have presented a novel bidirectional DCNN framework with Bayesian modelling for detection of features and landmarks, where the recurrent representation of an image is delivered through both forward and reverse branches to extract robust information of concrete features or salient objects. With the incorporation of the Gaussian spatial indexing module in the proposed network, the detected salient features can be further converted into coordinates for accurate landmark detection. Extensive experiments and thorough comparisons verify the high performance of the approach compared to the state-of-the-art models in crack detection and facial landmark detection.

References

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

[2] 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 *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recog.*, pp. 7268–7277, 2018.

[3] 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. 60201–60214, 2021.

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

[5] C. Finn, X. Y. Tan, Y. Duan, T. Darrell, S. Levine, and P. Abbeel, "Deep spatial autoencoders for visuomotor learning," in *2016 IEEE Int. Conf. Robot. Autom. (ICRA)*, pp. 512–519, IEEE, 2016.

[6] 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.

[7] 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 *Int. Joint Conf. Neural Netw.*, pp. 2039–2047, 2017.

[8] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S.-M. Hu, "Global contrast based salient region detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 569–582, 2015.

[9] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "300 faces in-the-wild challenge: The first facial landmark localization challenge," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recog. Workshops*, pp. 397–403, 2013.

[10] F. Yang, L. Zhang, S. Yu, D. Prokhorov, X. Mei, and H. Ling, "Feature pyramid and hierarchical boosting network for pavement crack detection," *IEEE Trans. Intell. Transp. Sys.*, vol. 21, no. 4, pp. 1525–1535, 2019.

[11] 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 Trans. Ind. Inform.*, vol. 16, no. 12, pp. 7448–7458, 2019.

[12] 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. Ind. Inform.*, 2020.

[13] 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.

[14] 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.