JCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-7, July 2018 E-ISSN: 2347-2693

Deep Belief Network and its application for Detection of Concrete Surface Cracks

Khalid Hussain

University of Kashmir, Srinagar, India

*Corresponding Author: khalid.hussain@uok.edu.in

Available online at: www.ijcseonline.org

Accepted: 17/Jul/2018, Published: 31/July/2018

Abstract—Safety inspection of concrete surfaces like road and bridge surfaces is a continuous and critical task since it is closely related with structural health and reliability of such surfaces. However, it is difficult to find cracks by visual check especially for large and complex concrete surfaces like roads and bridges. Automation in structural strength monitoring of concrete surfaces has generated a lot of interest in recent years, mainly because of introduction of cheap digital cameras and microcontrollers. However, it is still tough task because of the intensity homogeneity of cracks and complexity of the background. Inspired by recent success on applying deep learning to complex computer problems like vision, object detection etc., deep learning based algorithm is proposed in this paper for detection of cracks on concrete surfaces. The proposed algorithm uses Deep Belief Network (DBN), which is trained using an image data set of 600 crack images of concrete surfaces like bridges, roads etc. collected by low cost smart phones. By the analysis of experimental data, the algorithm successfully detects images with cracks of various types. The recognition accuracy is more than 88% compared with 70% accuracy from a typical image based approach. The results are also compared with SVM (Support Vector Machine) and traditional approaches and the recognition accuracy in DBN approach has been found much higher than in these approaches. This algorithm if implemented on a robotic device or simple vehicle with image acquisition capability can prove very beneficial for non-expert inspectors, enabling them to perform crack monitoring tasks efficiently.

Keywords: Deep Learning, Deep Belief Networks, Restricted Boltzmann Machine.

I. INTRODUCTION

All concrete surfaces including concrete surfaces of bridges and expressways are very vulnerable to cracks due to many factors like design, material, loads, and environmental conditions. The condition assessment of such surfaces plays a crucial role in maintaining their structural health, reliability and durability. The early detection of such cracks on concrete surfaces is a crucial and important maintenance task and can lead to prevention of accidents, and increasing life of surfaces. The current method of site inspection is very time-consuming process especially for long roads and bridges. The scanning of concrete surfaces of bridges etc. using robots is a new trend [1] but challenge lies in automated interpretation of large image data set in order to infer the condition of concrete surfaces. Most of the automatic crack detection methods use simple edge following, image thresholding and morphology operations [2-9] but these methods are not robust to noise and require manual parameter setting and adjustment. Further such algorithms work well only when cracks are high contrast regions with a near uniform background. However, in real world the concrete images have cracks of variable appearance with a lot of competing visual clutter as shown in figure 1. Further adjusting parameters manually is hectic task that too when size of data set is large. Recently computer vision and machine learning techniques have been successfully applied to automate and monitor concrete surfaces like roads for various conditions like strength etc. For the task of automated surface crack detection, the trend toward using deep learning algorithms is relatively new. As machine learning is a vast field and there exists no best algorithm for all classification tasks, to build any algorithm requires developing the right representation of the data. By proposing a deep learning based automatic crack detection system in this paper, it is possible to secure the stability of surface, reduce no of inspections, minimal lane closure, inspection time, and maintenance cost. Further the condition of structural health can be judged objectively by acquiring and processing data. Deep learning will rely on discriminative and representative deep features of training set.

Rest of paper is organized as follows. Section II contains the related work in the area of classification tasks especially done on concrete surface crack images using various techniques. Section III contains the framework of proposed work and various steps involved in the whole process of classification. Section IV explains various steps taken for pre-processing of images used for training. Section V explains deep belief networks and a training algorithm for deep belief network.

Section VI contains classification results of proposed algorithm as well as comparisons with other algorithms. Section VII contains future work which can be undertaken on the work proposed in this paper.

II. RELATED WORK

Chae *et al*. [10] proposed a neuro-fuzzy approach to determine the health of sanitary sewer pipelines. Pre-processed images were fed to neural networks and the network was able to identify various cracks and joints. Khanfar *et al*. (2003) [11] proposed a non-destructive test using wavelets for detecting cracks through changes in the reflection coefficient of a concrete surface. The model is based on fuzzy logic model which uses the reflection coefficient, frequency of actions and standoff distance to estimate the crack width and depth.

Hyeong-Gyeong Moon and Jung-Hoon Kim [12] proposed a simple neural network based approach with 5 hidden layers with the inputs being area and major to minor axes ratios of all the objects in an image at a time to determine the presence of cracks. The object properties are determined after image processing involving techniques of subtraction processing, gaussian filtering, thresholding, morphological closing, and labeling.

A visual technique [13] involves pre-processing of images using subtraction of smooth images and a hessian matrix based filter, which then detected cracks by applying a threshold.

Gajanan K. Choudhary and Sayan Dey proposed crack detection in concrete surface using image processing based on fuzzy logic and neural network [14]. They use feature extraction that is crack like regions detection based on edge detection procedure and then apply fuzzy and neural network approach to classify crack and noise. A support vector based method is proposed in [15,16] for detection and recognition of pavement distress in concrete surfaces.

Machine learning has been applied for visual recognition and classification and generally performs better than methods with hand-tuned parameters. Example based machine learning is referred to as supervised learning and enables statistical inference based on the relevant data without the need for manual parameter adjustment. In the recent years, effective algorithms for medical imaging and computer vision tasks were developed using deep neural networks [17-22] with high accuracy rate than conventional neural networks.

In this paper, deep learning based method is proposed which learns discriminative features directly from raw image patches using Deep Belief Networks. The method does not make any assumption of the geometry of the pavement as required in [23]. Further discriminative features are automatically learned from images rather than using hand-crafted features as in [23,24]. Further the proposed method can be successfully applied to images collected using low-cost smart phone even with a complex background.

Figure 1. Top row shows high contrast, low clutter images, Bottom row shows real surface cracks

III. PROPOSED METHOD

The proposed method for detection of cracks involves following three steps:

1. Pre-processing of images.
2. Building and Training deep belief network.
3. Testing deep belief network with a test data set for classification.

Data set of images needs to be pre-processed to remove any irrelevant information and un-necessary noise to facilitate efficient training of deep belief network and thus resulting in more accurate classification. The flowchart of whole process is shown in figure 2. The pre-processing is discussed in next section while as deep belief networks and their training procedure is discussed in section V.

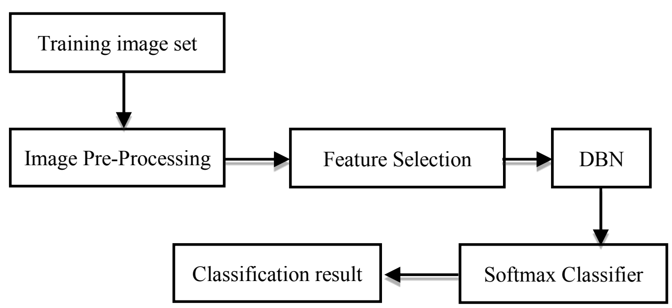

Figure 2: Steps of proposed algorithm

IV. PRE-PROCESSING OF IMAGE DATASET

Various operations which are performed on images include Resizing, RGB-Grayscale image conversion, Edge detection, Morphological operations, Connected component labeling and Identifying object properties [14].

A. Resizing and RGB-Grayscale conversion:
All the images are resized to a size of 28x28 pixels. This step provides uniformity and saves computational time at the cost of loss of some information in the image. The images are converted from an RGB image to grayscale after resizing them. This is necessary to eliminate the effects of most of the color differences in the images.

B. Edge detection:
In this phase, edges are detected based on calculation of gradient in the image. This operation detects an edge wherever there is a steep gradient due to change in color or luminance.

C. Morphological operations:
Various morphological operations are performed on the detected edges for reducing more noise and for joining fragmented cracks. Various steps involve close, bridge, spur and clean. Close is the process, by which small holes in the objects are filled, it involves dilation followed by erosion. Bridge is used for joining a fragmented crack which arises due to earlier edge detection process. Bridge may lead to joining of two nearby objects which may be noise but the pros of joining fragmented cracks are much more than addition of noisy cracks. Spur operation removes individual pixels that are only diagonally connected at the border of an object in a binary image, called spur pixels. Clean operation converts isolated 'on' pixels to zero. These objects are the easiest to classify as noise, so this step has been used only because it helps in noise reduction, and thus reduces the number of objects to be classified in image and thus resulting in saving computational time.

D. Labeling of connected component:
This performs serial numbering of all connected regions of an image. It helps in identification of continuous, connected regions and assists in identifying object properties.

E. Identification of Object propertie:
Identifying object properties form the basis for crack detection using neural networks. The properties should be chosen in such a way that there should be clear differentiation between noise and cracks. The two properties which are chosen are Area and Major-Minor axes ratio. Area of object in an image is equal to total no of pixels that make up the object. This property is chosen because area of cracks is higher than noise. The major and minor axes are the longest dimension of any object and the longest dimension perpendicular to the major axis at its midpoint such that the ellipse of these two dimensions as major and minor axes completely contains the given object. Usually major-minor axis ratio is higher for cracks as they are elongated rather than noise. The pre-processed data set of images is then used for training DBN. The result of various operations on a sample image from a dataset in shown in figure 3:

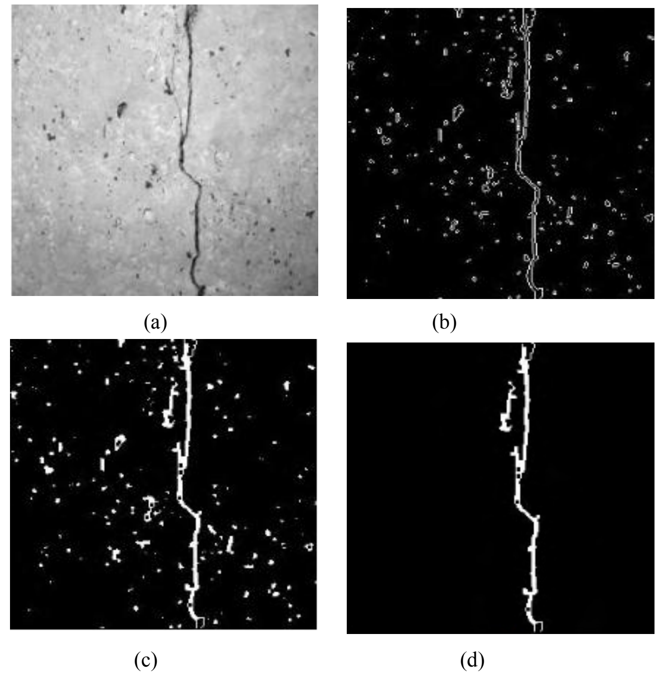

Figure 3. (a) Original image converted into grayscale image.
(b) Feature extraction using sobel edge detection.
(c) Labeled image.
(d) Detected crack at the end of image pre-processing phase.

V. DEEP BELIEF NETWORK MODELLING

The proposed method uses Deep Belief Networks (DBN's) for building classification model and feature learning. Deep belief networks can be trained on large unlabeled data and a small amount of labeled data to classify crack images by extracting core features from underlying features. Deep belief network belongs to class of deep neural networks composed of multiple layers of hidden units that are interconnected but there are no connections between units within each layer. DBN is able to probabilistically reconstruct their inputs when they are trained in unsupervised mode on a training set. The layers then act as feature detectors. After this learning step, a DBN can be further trained with supervision to perform classification. The basic structure of DBN is shown in figure 4:

A. Restricted Boltzmann machine theory:
The DBN used in this work is composed of Multi-layer Unsupervised Restricted Boltzmann Machine (RBM) network and a layer of supervised back-propagation. It is two layer, undirected and bipartite graphical model [25]. The first layer consists of visible units or observed data variables and the second layer consists of hidden units or latent variables.
Both these layers are fully connected via symmetric undirected weights without any intra-layer connections within each of the layers. RBM model topology is shown in figure 5.

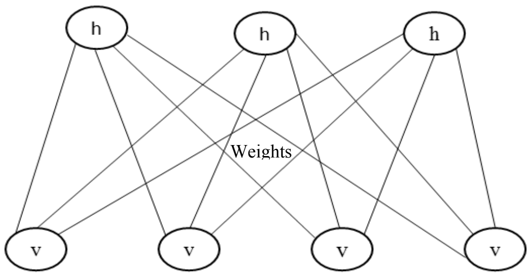

Figure 4: 'h' nodes represent hidden layer, 'v' nodes represent visible layer, no interconnections among units within layer

The most important advantages of RBM is that the activation state of each of hidden units and visible units are conditionally independent when given the state of visible units and hidden units respectively. The layers are trained efficiently by using the feature activations of one layer as the training data for the next layer. Better initial values of weights in all layers can be obtained by the layer-wise unsupervised training, compared to random initialization. RBM's have been used as generative models of many different types of data including labeled or unlabeled images, bags of words that represent documents, and user ratings of movies. In their conditional form they can be used to model high-dimensional temporal sequences such as video or motion capture data or speech. Their most important use is as learning modules that are composed to form deep belief nets.

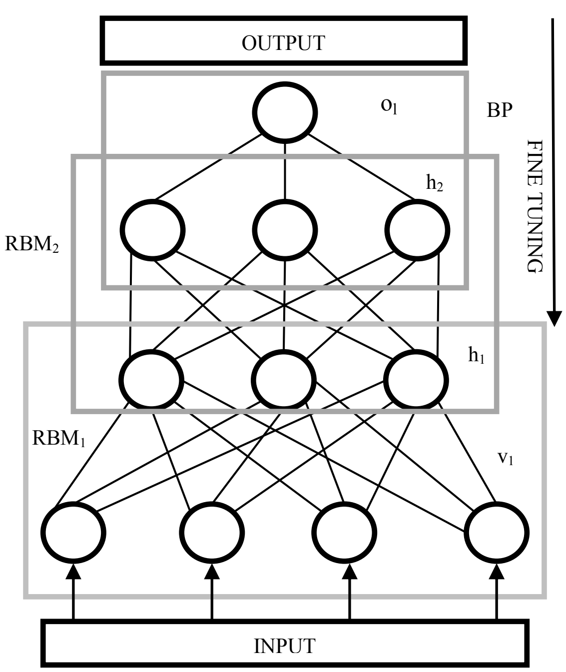

Figure 5. Architecture of DBN with two RBM's. 'v_l' represents visible layer, h_1 represents hidden layer 1, h_2 represents hidden layer 2, 'o_l' represents output layer, BP: Back propagation.

B. RBM energy model:

The Boltzmann machine is a type of Markow random field (MRF). It is a concurrent network with stochastic binary units. It is an energy based model and the energy function of RBM is given by equation 1:

$$E(v,h \mid \theta) = -va^T - hb^T - wv^T h$$

$$= -\sum_{p=1}^{n} v_p a_p - \sum_{q=1}^{m} h_q b_q - \sum_{p=1}^{n}\sum_{q=1}^{m} w_{pq} v_p h_q \qquad (1)$$

Where,

n= number of visible layer units, m= number of hidden layer units.

v_p=p_{th} unit of the visible layer h_q=q_{th} unit of the hidden layer.

a_p=bias weight of visible units b_q=bias weight of hidden units.

w_{pq}= symmetric interaction term between visible unit p and hidden unit q. θ=(W_{pq} , a_p , b_q) represents all the parameters of RBM.

The joint probability distribution for visible and hidden units is defined in terms of energy function and shown in equation 2:

$$P(v,h) = e^{-E(v,h)} / H(\theta) \qquad (2)$$

where

$$H(\theta) = \sum_{v,h} e^{-E(v,h \mid 6)} \qquad (3)$$

H(θ) is normalization factor also called a partition function (equation 3). It acts as a normalizing constant, which ensures that probability distribution sums to 0. As there are no intra layer connections, it makes the first expectation easy to compute [26]. Given a visible vector, the activation state of each of the hidden units are conditionally independent. Given a visible vector the activation probability of j_{th} hidden unit is given by equation 4:

$$P(h_q = 1 \mid v) = \sigma(\sum_{p=1}^{n} W_{pq} v_p + a_q) \qquad (4)$$

$$\sigma(x) = 1/1 + e^{-x} \qquad (5)$$

Where $\sigma(x)$ is a sigmoid activation function (equation 5) whose range is usually (0 to 1). In the same way, unbiased sample of the state of visible unit can be found given a hidden vector and shown in equation 6.

$$P(v_p = 1 \mid h) = \sigma(\sum_{q=1}^{m} W_{pq} h_q + b_p) \qquad (6)$$

C. DBN Training process:

RBM learning process is usually unsupervised learning, so if DBN is to be used for classification, a new network of supervised learning has to be added which can classify concrete surface images based on features extracted by DBN. The overall training process can be divided into two phases: Pre-Training phase or Layer wise unsupervised learning and Fine tuning Phase.

Pre-Training phase:
The pre-training corresponds to an efficient learning technique that stacks RBMs, which are independently trained layer-by-layer. RBM tries to adjust its parameters in such a way such that the probability distribution represented by it fits the distribution of the training data. Initially RBM is trained by inputting the original data and fixing up the parameters of first RBM, then output of first RBM is fed as input to second RBM and so on. This method of training an RBM can be repeated several times to create a multilayer model whose parameters are suitable to extract the features of training data [27]. The detailed algorithm in steps is discussed in later part of section.

Fine-Tuning Phase:
In this phase, supervised training is used that fine tunes all layers jointly to perform the classification task. This fine-tuning is done by initiating the parameters of a deep neural network with the values of pre-trained DBN parameters. After that, a final layer composed of two softmax units is used to perform a binary classification. Further the back propagation algorithm and a gradient-based optimisation algorithm can be used to adjust the network parameters, creating a DBN based Deep Neural Network.

D. DBN Training algorithm:
The algorithm used to train DBN in this work is based on contrastive divergence (CD) proposed by Hinton [26] and commonly used algorithm to train DBN's. It is an approximation of the log-likelihood gradient and has been found to be a successful update rule for training RBM. The step by step procedure of training using CD algorithm is given below:

i. Initialization:
Initialize the state of visible units with training sample: $v_1=x_0$. Initialize w, a, b to set of small random initial values.

ii. Hidden layer update:
Determine the state of all the units of hidden layer h_1 through visible layer v_1 with the help of equation 4.

iii. Visible layer update:
Determine the state of visible layer v_2 through hidden layer h_1 with the help of equation 6.

iv. Hidden layer update:
Determine the state of hidden layer h_2 through visible layer v_2 with the help of equation 4.

v. Parameter Updation:
Update parameter (W, a , b) in order make it to fit a training data. Following the gradient of the log likelihood logp(v) we obtain the update rule for the weights as:

$$\Delta w_{pq} = \eta\left(< v_p h_q >_{data} - < v_p h_q >_{model}\right)$$
$$\Delta a_p = \eta\left(< v_p >_{data} - < v_p >_{model}\right)$$
$$\Delta b_q = \eta\left(< h_q >_{data} - < h_q >_{model}\right)$$

where η is learning rate and $< v_p h_q >_{data}$ is the expectation observed in the training set of images and $< v_p h_q >_{model}$ is the same expectation under the distribution defined by model. However, Contrastive Divergence (CD) approximation to the gradient is used where $< v_p h_q >_{model}$ is replaced by running the Gibbs sampler initialized at the date for one full step. [28][29]

vi. Repeat:
Get next sample of data and repeat above steps (step ii to step v). Once an RBM is trained, another RBM is stacked atop it, taking its input from the final trained layer. The new visible layer is initialized to a training vector, and values for the units in the already trained layers are assigned using the current weights and biases. The new RBM is then trained with the procedure above. This whole process is repeated until the desired stopping criterion is met.

VI. RESULTS

In order to evaluate the performance and usability of the proposed algorithm, data set comprising of 200 images normalized to size of 28*28 of each type (longitudinal, transverse, and network) captured by low cost smart phone and pre-processed initially are used for training while as 50 crack images of each type are used for testing. The experimental results are shown in table 1, 2 and 3. The DBN is composed of five layers, the input layer is 28*28 (784 neurons). The hidden layer is composed of 1000-1000-1000 neurons with unsupervised learning rate of 0.01 and supervised learning rate of 0.1:

Table 1. Recognition accuracy of transverse crack test images

Transverse crack images	
Training Set: 200 images	Test Set: 50 images
No. of cracks correctly detected	44 out of 50
Recognition accuracy	88 %

Table 2. Recognition accuracy of longitudinal crack test images

Longitudinal crack images	
Training Set: 200 images	Test Set: 50 images
No. of cracks correctly detected	45 out of 50
Recognition accuracy	90%

Table 3. Recognition accuracy of network crack test images

Network crack images Training Set: 200 images	Test Set: 50 images
No. of cracks correctly detected	43 out of 50
Recognition accuracy	86%

As is evident from Table 1, 2, and 3 the classification and recognition accuracy of the proposed DBN based method for three major types of cracks is more than 88%. For transverse and longitudinal cracks the accuracy is 88% and 90% respectively while as for network cracks it is only 86% due to the complexity of network cracks. The recognition accuracy of non-crack image is found to be 91%. The results are also compared with SVM (Support Vector Machine) method trained with LIBSVM [21]. The DBN based approach performed well with an average recognition accuracy of 88% while as SVM and Boosting trained on same data set showed recognition accuracy much lower than proposed DBN based approach. The results are shown in table 4.

Table 4. Comparison of SVM, Boosting and DBN

Classifier	**Recognition accuracy**
Boosting	79%
SVM	81%
DBN	88%

Accuracy of DBN classification algorithm is much higher than SVM and other traditional approaches because DBN can extract abstract features which can greatly improve the performance of the classifier. The main cause of recognition errors is the uneven light, number of cracks, image tilt angle and so on.

VII. CONCLUSION

In this paper, an algorithm based on Deep Belief Network is proposed for automated detection of surface cracks. It initially uses various image processing techniques to enhance the quality of data set. Finally, Deep Belief Network is able to classify the images with different types of cracks. The experimental data show that this method can effectively determine the types of cracks and the classification accuracy can meet the requirements of engineering best practices. The algorithm needs to be optimized for improving accuracy especially when the test images are collected from different geographical area with different weather, light and road conditions. In addition, the various optimization methods can be applied to determine optimal parameters required in image processing. Further the proposed need to be compared with other algorithms on the basis of various criteria like precision, sensitivity, recall, specificity etc. to measure its accurate performance.

REFERENCES

[1] H. M. La, R. S. Lim, B. B. Basily, N. Gucunski, J. Yi, A. Maher, F. A. Romero, and H. Parvardeh, “Mechatronic systems design for an autonomous robotic system for high-efficiency bridge deck inspection and evaluation,” IEEE/ASME Trans.

[2] T. Nishikawa, J. Yoshida, T. Sugiyama, and Y. Fujino, “Concrete crack detection by multiple sequential image filtering,” Comput. Aided Civil Infrastructure Eng., vol. 27, no. 1, pp. 29–47, 2012.

[3] Z. Zhu, S. German, and I. Brilakis, “Visual retrieval of concrete crack properties for automated post-earthquake structural safety evaluation,” Autom. Construction, vol. 20, no. 7, pp. 874–883, 2011.

[4] T. Yamaguchi and S. Hashimoto, “Fast crack detection method for large-size concrete surface images using percolation based image processing,” Mach. Vision Appl., vol. 21, no. 5, pp. 797–809, 2010.

[5] T. Yamaguchi, S. Nakamura, and S. Hashimoto, “An efficient crack detection method using percolation-based image processing,” in Proc. 3rd IEEE Conf. Ind. Electron. Appl., Jun. 2008, pp. 1875–1880.

[6] X. Tong, J. Guo, Y. Ling, and Z. Yin, “A new image based method for concrete bridge bottom crack detection,” in Proc. Int. Conf. Image Anal. Signal Process. (IASP), Oct. 2011, pp. 568–571.

[7] H.N. Nguyen, T.Y. Kam, and P.Y. Cheng, “A novel automatic concrete surface crack identification using isotropic undecimated wavelet transform,” in Proc. Int. Symp. Intell. Signal Process. Commun. Syst. (ISPACS), Nov. 2012, pp. 766–771.

[8] R. Adhikari, O. Moselhi, and A. Bagchi, “Image-based retrieval of concrete crack properties for bridge inspection,” Autom. Construction, vol. 39, pp. 180–194, 2014.

[9] R. S. Lim, H. M. La, Z. Shan, and W. Sheng, “Developing a crack inspection robot for bridge maintenance,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), May 2011, pp. 6288–6293.

[10] Chae M., and Abraham D., (2001). ”Neuro-Fuzzy Approaches for Sanitary Sewer Pipeline Condition Assessment.” Journal of Computing in Civil Engineering 15, Special issue: Information technology for life-cycle infrastructure management, Pages 4–14. doi: 10.1061/(ASCE)0887-3801(2001)15:1(4).

[11] Aws Khanfar, Mohammed Abu-Khousa, and Nasser Qaddoumi, “Microwave near-field non-destructive detection and characterization of dis-bonds in concrete structures using fuzzy logic techniques,” Composite Structures, Volume 62, Issues 3–4, Pages 335-339, ISSN 0263-8223, 10.1016/j.compstruct. 2003. 09.033, 2003

[12] H. Moon, and J. Kim, “Intelligent Crack Detecting Algorithm On The Concrete Crack Image Using Neural network,” Proceedings of the 28th ISARC, Pages 1461-1467, Seoul, Korea, 2011.

[13] Y. Fujita, Y. Mitani and Y. Hamamoto, “A Method for Crack detection on a Concrete Structure,” 18th International Conference on Pattern Recognition, Volume 3, pp. 901–904, 2006.

[14] Gajanan K. Choudhary and Sayan Dey. Crack Detection in Concrete Surfaces using Image Processing, Fuzzy Logic, and Neural Networks‖. 2012 IEEE fifth International Conference on Advanced Computational Intelligence (ICACI) October 18-20, 2012.

[15] DING Ailing, JIAO Licheng. Pavement distress recognition based on support vector machine [J]. Journal of Changan

University: Natural Science Edition, 2007, 27(2):34-37.

[16] Yan H S, Xu D. An Approach to Estimating Product Design Time Based on Fuzzy Support Vector Machine [J]. Neural Networks IEEE Transactions on, 2007, 18(3):721-731.

[17] H. Roth, L. Lu, J. Liu, J. Yao, A. Seff, C. Kevin, L. Kim, and R. Summers, “Improving computer-aided detection using convolutional neural networks and random view aggregation,” IEEE Transactions on Medical Imaging, 2015.

[18] D. Ciresan, A. Giusti, L. M Gambardella, and J. Schmidhuber, “Deep neural networks segment neuronal membranes in electron microscopy images,” in Advances in Neural Information Processing Systems, 2012, pp. 2843–2851

[19] D. C Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, “Mitosis detection in breast cancer histology images with deep neural networks,” in Medical Image Computing and Computer-Assisted Intervention (MICCAI), pp. 411–418. 2013

[20] A. Krizhevsky, I. Sutskever, and G. Hinton, “Imagenet classification with deep convolutional neural networks,” in Advances in Neural information Processing Systems, 2012, pp. 1097–1105

[21] Y. Zhang, K. Sohn, R. Villegas, G. Pan, and H. Lee, “Improving object detection with deep convolutional networks via Bayesian optimization and structured prediction,” in Proceeding of IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 249–258.

[22] J.J. Kivinen, C. K. Williams, and N. Heess, “Visual boundary prediction: A deep neural prediction network and quality dissection,” in Proceedings of International Conference on Artificial Intelligence and Statistics, 2014, pp. 512–521.

[23] S. Varadharajan, S. Jose, K. Sharma, L. Wander, and C. Mertz, “Vision for road inspection,” in Proceedings of 2014 IEEE Winter Conference on Applications of Computer Vision, 2014, pp. 115122.

[24] Y. Hu and C. Zhao, “A local binary pattern based methods for pavement crack detection,” Journal of Pattern Recognition Research, vol. 5, no. 1, pp. 140–147, 2010.

[25] C. Zhang, N. Ji, and G. Wang, Restricted Boltzmann Machines, Chinese Journal of Engineering Mathematics [J], 2015,(2):159-173.

[26] G. Hinton, "A practical guide to training restricted boltzmann machines," Machine Learning Group, University of Toronto, Technical report, 2010.

[27] Hinton G E. Training products of experts by minimizing contrastive divergence [J]. Neural computation, 2002, 14(8): 1771-1800.

[28] Hinton, G., Osindero, S., and Teh, Y. “A fast learning algorithm for deep belief nets,” Neural Computation, vol. 18, 2006, pp. 1527-1554

Author Profile

Mr. Khalid Hussain has pursued his B.Tech and M.Tech in Computer Science and Engineering from GGS Indraprastha University, Delhi in year 2008 and 2011 respectively. He has formerly worked as Assistant Professor in ASET-Amity University Haryana. He is presently working as Assistant Professor in Department of Computer Science and Engineering, University of Kashmir. His main research work focuses on Artificial intelligence, Machine learning, Data Analytics, Data Mining and Theoretical Computer Science.

