P-ISSN 2586-9000
E-ISSN 2586-9027
Homepage : https://tci-thaijo.org/index.php/SciTechAsia
Science & Technology Asia
Vol. 23 No.2 April - June 2018
Page: [19 – 28]
Original research article

Concrete Crack Detection Using the Integration of Convolutional Neural Network and Support Vector Machine

Mayank Sharma[1,*] Weerachai Anotaipaiboon[1]
[1]*Faculty of Engineering, Department of Electrical and Computer Science, Thammasat University, Pathum Thani 12120, Thailand*

Krisada Chaiyasarn[2]
[2]*Faculty of Engineering, Department of Civil Engineering, Thammasat University, Pathum Thani 12120, Thailand*
Received 11 September 2017; Received in revised form 9 February 2018
Accepted 19 February 2018; Available online 30 June 2018

ABSTRACT

Crack detection in concrete structures is an important task in the inspection of buildings to ensure their safety and serviceability. Previous studies relating to crack detection via image-processing and machine learning techniques generally involve the complex modelling of cracks and the extraction of hand crafted crack features. This approach often fails to apply to images from a real environment. This paper proposes a new image-based crack detection system using a combined model of classifiers, namely a Convolutional Neural Network (CNN) and a Support Vector Machine (SVM), which was proven to perform better than the methods involving the handcrafted features. In the proposed technique, a CNN is used in extracting deep convolutional features from the RGB images of cracks and an SVM classifier is used as an alternative to a softmax layer to enhance classification ability. The combined model automatically extracts features and determines whether or not an image patch belongs to a crack class. A dataset of 550 images are collected by a digital camera from various locations, and from the results, it is concluded that the proposed method is able to identify cracks on concrete surface with an accuracy of 90.76 %

Keywords: Convolutional Neural Network; Computer Vision; Concrete Structure; Crack Detection; Support Vector Machine

1. Introduction

As concrete structures age, regular assessment and maintenance work is essential to obtain information about the structural condition. The most common inspection procedure is visual inspection, which is used to assess the physical and functional condition of structures. Visual

*****Corresponding author**: macx.sharma@gmail.com
doi: 10.14456/scitechasia.2018.11

inspection is usually carried out by experienced engineers to detect and monitor defects within the aging structures. This procedure is costly and labor intensive. In visual inspection, crack detection is an important procedure as cracks provide significant information about the current state of a concrete structure as discussed in [1, 2, 3]. Fig.1 shows some example images of cracks on concrete surfaces used in this study. A number of previous studies have been conducted in developing automatic crack detection systems, although much work still is required if they are to be adopted in practice. In crack detection, pattern recognition is applied to obtain the prominent features of cracks from digital images or video data. Previous methods of crack detection applied handcrafted features extraction techniques to obtain unique crack features from images, which can be complex and time-consuming, unlike techniques using a Convolutional Neural Network (CNN), which can retrieve semantic information from images. As exemplified in [4] in handwritten digit recognition, CNN was applied to automatically extract optimized features for a Support Vector Machine (SVM) to classify, which shows better recognition results. This paper presents a similar approach by combining CNN and SVM for crack detection. The aim is to develop a new and reliable method based on the state-of-art computer vision technology to improve accuracy and efficiency in automatic crack detection. In this study, our primary focus is on detecting the presence of cracks in the images, which can be applied to assess concrete structures for building inspection.

Fig. 1. Example images of cracks used in this study

This paper is organized as follows; the background study is included in section II. An overview of the proposed system is explained in section III and in section IV, implementation detail is provided. Lastly, results and discussion are provided in section V, followed by the conclusion.

2. Background

2.1 Image-based crack detection

In previous studies, crack detection systems are based on the assumption that crack pixels have lower intensity than background pixels; hence, thresholding techniques can be applied to extract crack regions in images. The crack detection system as shown in [5] involves two steps: (i) a line filer based on the hessian matrix to enhance the line features associated with cracks, and (ii) thresholding to extract crack regions. In [6], an image-based crack detection method was proposed to automate crack detection in bridges. In this paper, an image division operation was applied, thus splitting an image into two different regions and the regions were then classified to be either crack or non-crack based on the region features. However, threshold-based algorithms are prone to errors caused by shadow. In [7], a percolation-based system was proposed for pavement crack detection by adopting a template matching technique, which can reduce the system computational cost without affecting the results of crack detection. The modeling process starts by initializing a seed region and then the labelling operation will be executed on neighbouring regions as crack regions, based on the percolation process [8]. Amhaz et. al. [9] developed the algorithm based on Minimal Path Selection (MPS) for automatic crack detection by adding the width estimation. The model-based approach often relies on the user input to initialize the seed pixels.

Crack detection algorithms based on pre-processing have been proposed to facilitate the laborious process of visual inspection. In [10], various image-based edge detection techniques for detecting cracks within concrete structures were compared against the Wavelet, Fourier Transforms, Sobel, and canny algorithms. It was observed that the fast Haar wavelet-based method was more reliable than the other methods. However, in the case of noisy blemishes in images, edge detection algorithms show a dramatic drop in their performance. For extracting significant crack features from images, a principle component analysis (PCA) algorithm was applied to mitigate the dimensionality problem of feature vectors [11]. Then for the classification step, the nearest neighbour algorithm was used in this system. However, in the case of noisy blemishes images, methods based on pre-processing can fall short in their performance due to the level of image noise.

2.2 Crack detection based on machine learning

To this end, algorithms that classify cracks using machine learning have been proposed. SVM is a supervised learning technique in machine learning invented in 1995 by Vapnik and Cortes [12]. Lui et. al. [13] applied a SVM classifier to determine if cracks appear in images, in which the images are pre-processed to extract crack features based on pixel-intensity. A crack detection algorithm using neural network was explained in [14] with five hidden layers. The given inputs are the ratio of the major and minor axis rotation and the area of an object. The features are generated from various operations, such as subtraction processing, Gaussian filtering, morphological closing and thresholding. A preliminary study was conducted in [15] to identify and classify cracks using texture-based feature extraction. In [16], a minimum spanning tree (MSTs) was used to extract the possible connections of sampled crack seeds and undesired edges in trees that are then pruned to obtain crack curves. Parsana et. al. [17] developed a vision-based automatic crack detection system to detect cracks on a bridge deck. The detection system starts by constructing feature vectors based on intensities gradient and scale-space. The feature vectors are then classified using various classifiers. Shi et. al. [18] developed an automated road crack detection method by extracting crack features based on a discriminitive integral channel and then classifying the features by random structure forests. This method can deal with crack intensity inhomogeneity, which can capture and utilise some unique characteristics of cracks.

The Convolutional Neural Network (CNN) has recently been applied in visual computing due to its promising results [19, 20, 21, 22, 23, 24, 25]. The work of [26] developed a CNN-based defect detector system for tunnel inspection. The inspection system starts by extracting feature vectors of defects based on different edge, frequency, texture, entropy, scale invariance and HOG. The feature vectors are classified using CNN, and then compared with various classifiers. Although this method shows staisfactory results, the method has to be carried out off-line for training and classifying as there are too many steps in the features extraction stage. Zhang et. al. [27] developed a CNN-based road crack detection method by automatically extracting the discriminative features from RGB images at multiple levels, without the need of hand-crafted features, and then applying the CNN softmax layer to classify features. As shown in previous studies, the multi-level deep features learned by the CNN-based methods will likely replace the hand-crafted features-based techniques [20]. Such promising results motivated the use of the CNN-based technique to solve the crack detection problem in this paper.

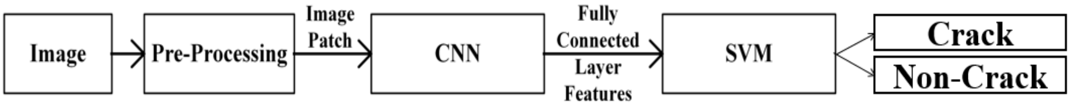

Fig. 2. System Pipeline

3. Methodology

3.1 Data Collection

In the proposed system, image-based visual inspection needs to collect data of cracks on different concrete surfaces. As shown in Fig. 2, the first process is image acquisition, in which images of concrete surface cracks have been taken by a digital camera to build a dataset of crack and non-crack images from various locations, mainly inside the Thammasat University campus. For training, a CNN classifier, image patches are used instead of full images. Training and validation images were collected in various locations for different types of concrete surfaces, and the images were manually labelled to create a database of cracks. These images were used in classification as well as in validation.

3.2 Pre-processing step

The image pre-processing technique is applied to obtain image patches using the Adobe® Photoshop software package, which divides RGB images into image patches with the size of 28x28 pixels. The patches are then used for training the CNN. Before feeding the CNN with training images, a labeling operation needs to be performed on the images. The training patches are labelled either one or zero, one is assigned to a patch containing cracks and zero is assigned to a non-crack patch. The RGB values are used as features in input vectors to the CNN.

3.3 Convolutional Neural Network

A CNN is a type of multilayer feed forward, biologically inspired, or an influenced variant of an artificial neural network. The CNN consists of three types of layers as shown in Fig. 3. A convolutional layer consists of a rectangular grid of filters, each of which takes input from rectangular regions of the input layer. Another layer of the CNN is max pooling, which is the form of non-linear down-sampling. Max pooling performs down-sampling operations to partition an input image into a set of small non-overlapping rectangular blocks. For each sub-region output block, a maximum value of the sub-region is computed as shown in Fig. 4 and a fully connected layer is used in computing each class score. In the proposed model, the Keras sequential model [28] is used in the proposed CNN architecture. As shown in Fig. 5, in the first two stages of convolution layer, activation layer (ReLU) and max-pooling layer are stacked. For the CNN, this is a suitable layer pattern for larger and deeper networks since multi-stacked convolutional layers can develop more complex features of the input data before the destructive pooling operation.

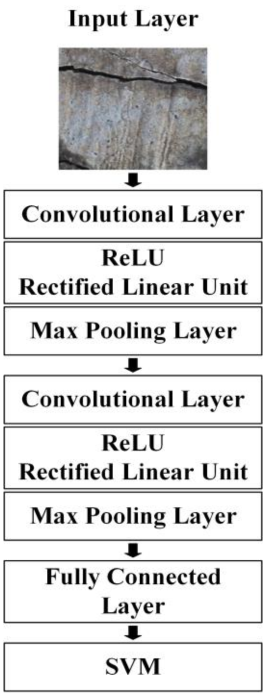

Fig. 3. The system pipeline with CNN layers architecture

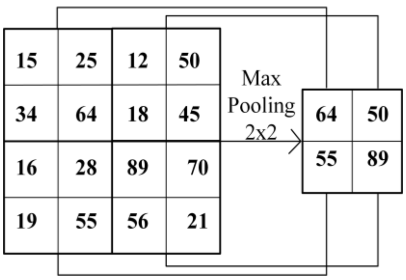

Fig. 4. An example of the Max pooling layer operation

In the last step, fully connected layer features are used for classification. Usually, a softmax classifier is used as the output layer of a convolutional neural network to compute the probability score of each class given an input patch. However, in the proposed model, the classifying step is replaced by a SVM classifier.

3.3.1 Convolutional Neural Network Learning

The main objective of training a convolutional network is to increase the variation of features and to avoid overfitting [27]. For training, the dropout method is used to reduce overfitting [29]. The training of the CNN is accelerated by using rectified linear units (ReLU) as the activation function [19]. The input to the CNN architecture is $r \times r \times d$ feature vectors, where r is the height and width of an image patch and $d = 3$, which is the RGB colour channels. The input training dataset is $\{I_c, y_c\}$, $c = 1, 2, 3, \ldots \ldots, n$, and n is the number of image patches used for training. A 28x28 pixels image patch, I_c, is the observation for the c^{th} image patch, and $y_c \in \{1, 0\}$ is the class label for the c^{th} image patch. Let G^k be the k^{th} output of the convolutional layer whose filters are determined by the weight W^k and bias b^k. Then the G^k is calculated as

$$G_{ij}^k = f\left(\left(W^k * x\right)_{ij} + b^k\right) \tag{3.1}$$

where x is the input of the convolutional layer and indices i and j correspond to the location of the input region where the filter is applied. Symbol $(*)$ and $f(\cdot)$ denoted the convolutional operator and non-linear function, respectively.

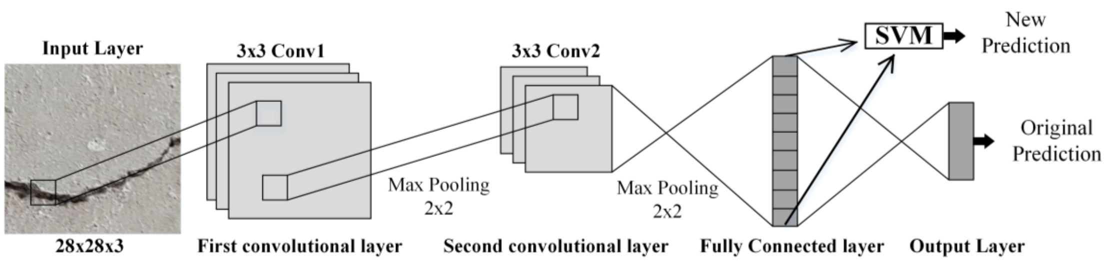

Fig. 5. Struture of the proposed CNN-SVM model

The first convolutional layer with 32 convolutional filters of size $3 \, x \, 3$ (mxm) pixels, where $m < r$, and the max pooling operation after the filtering has a ratio of 2. The CNN training procedure was stopped after 20 Epoch as it converged to a fixed value as shown in Fig. 6.

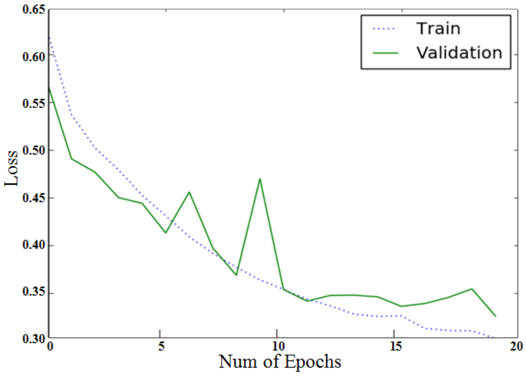

Fig. 6. A plot between Loss and Number of Epochs

3.4 Support Vector Machine

The idea behind a SVM is to find an optimal linear hyper plane decision boundary such that the expected classification error for testing samples is minimized. The working implementation of SVM classifier is summarized in detail in [23] . An SVM finds a hyperplane that separates the largest fraction of a labelled data set for binary classification, the training data is a set of training sample pairs $\{(x_1, y_1), \ldots, (x_i, y_i)\}$, where x_i is the observation for the i^{th} sample and $y_i \in \{1, 0\}$ is the associated class label. The SVM classifier is a discriminant function mapping an input vector space x into a class label.

$$f(x) = (w.x) + b \qquad (3.2)$$

where w is the weight or the direction of the linear decision boundary, and b is an added bias, which maximizes a margin between each class. In the proposed system, the CNN fully connected layer output features are the input for the SVM as depicted in Fig. 5, and a Radial basis (RBF) kernel is used. To obtain the optimal values for the kernel C and gamma, a cross-validation technique is employed. Different values of C and gamma have been experimented as shown in Table I. As shown in the table C = 4 and gamma = 1 provides the best accuracy.

Table 1. A Parametric study for support vector machine

Input Parameters		
C	gamma	Accuracy
1	0.5	89.26
1	1	89.48
1	2	86.05
1	3	83.17
2	1	90.60
3	1	90.62
4	**1**	**90.76**
5	1	90.67
6	1	90.64
7	1	90.57

4. Implementation

In this study, the proposed model employed CNN combined with SVM for crack detection. The CNN can be considered as the composition of two tools: an automatic multiple level feature extractor and a classifier. The feature extractor uses fully connected layers of the CNN to retrieve the information in the form of discriminative features from raw RGB images. The extracted features are used to train the classifier and the weight by a back-propagation algorithm. In the proposed model, a softmax classifier is replaced by a SVM classifier. In this study, the Scikit-learn toolbox [30] was used to build the SVM classifier in the experiment. In the experiment, the classifier not only predicts the output class labels, but also provides probability scores. The probability scores are then used to create the ROC shown in section 5.1. For evaluation of the proposed model, 15600 crack and non-crack image patches of different concrete surface types were used as depicted in Fig 7. The image patches were split into three sets, *i.e.* training, testing and validation data, with a split ratio of 8:2:2. For training 60% of image samples were selected randomly, 20% for validation and 20% for testing. The number of crack and non-crack image patches was set equally in each of the three datasets. The experiment was executed on a Window PC with Intel Core i7-4790 CPU with 8GB RAM using the Keras library implemented in Python.

(a)

(b)

Fig. 7. (a) Sample crack image patches (b) non-crack image patches

5. Results

The proposed CNN-SVM crack detection technique was evaluated using the ROC, which shows that the CNN-SVM performs well on validation images as shown in Fig. 9. Table II shows the results of the proposed technique, which show that the CNN-SVM model has the accuracy of 90.76%. It is clear that the combined model outper-forms the CNN model. Some image patches have been misclassified because the visibility of the crack is not clear in the image patches as shown in Fig 8. The overall efficiency of the crack detection system is promising.

Table 2. Performance evaluation of the proposed approach

Method	Accuracy
CNN	87.63 %
CNN-SVM	90.76 %

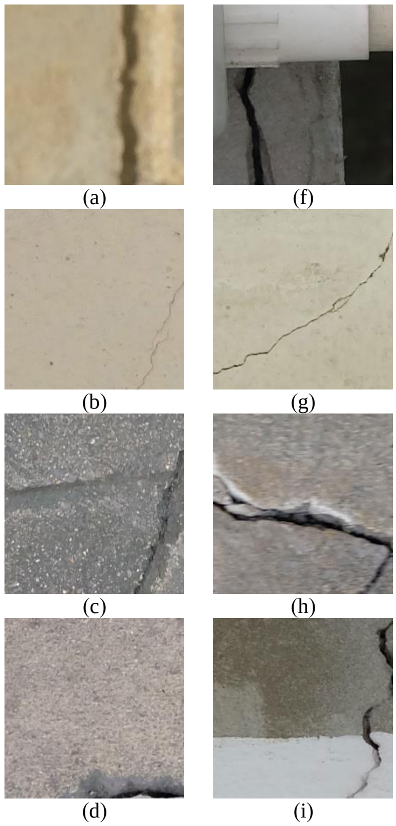

Fig. 8. Sample patches after being classified, incorrectly classified patches (a, b, c, d) and correctly classified image patches (f, g, h, i).

5.1 Receiver Operating Characteristic

The performance of the proposed CNN-SVM model was evaluated using the ROC curve, which is a plot between the true positive rate (TPR) and false positive rate (FPR) for different probability values of the output as computed by comparing predicted labels to ground truth values. As shown in Fig. 9, the true positive (TP), true negative (TN), false positive (FP), and false negative (FN) are computed from the frequency of occurrences when comparing between ground truth labels against predicted output labels, as shown in Table III. The best results can be achieved with CNN-SVM approach, as shown in Fig. 9.

Table 3. ConfusionMatrix for class classification.

Ground Truth Label	Predicted Label	
	Positive Prediction (Crack)	**Negative Prediction (Non-Crack)**
Positive Class (Crack)	True Positive (TP)	False Negative (FN)
Negative Class (Non-Crack)	False Positive (FP)	True Negative (TN)

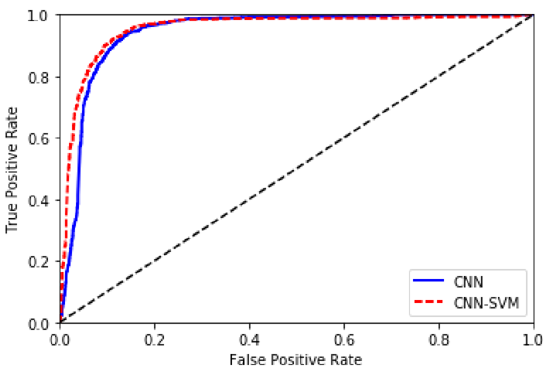

Fig. 9. The ROC curves for between the CNN technique, and the CNN-SVM technique.

6. Discussion

A previous researcher (Niu et al. (2011) [16] proved that better classification performance can be achieved with CNN features by using an SVM as a classifier. The proposed model of CNN-SVM slightly boosted classification accuracy by 87.63% to 90.76%. The increment is due to the use of a different optimization strategy. According to this author, the softmax layer classification performance is based on empirical risk minimization, which is required to minimize the predict-tion loss on the training dataset. On the other hand, the SVM is to find an optimal linear hyper plane decision boundary such that the expected classification error for testing samples is minimized, *i. e.* good generalization in performance, which requires minimizing the generalization error by using structural risk minimization principle. To maximize the margin results, the SVM provides better generalization ability compared to using the softmax as a classifier. Modifying the softmax classifier of the CNN with the SVM is simple, and appears to be useful for the crack detection problem. The CNN is capable of extracting discriminative features from a large amount of images without any pre-processing. Further work involves training and testing our system on a large database. For the inspection of a small section of concrete infrastructure, a hand held camera is sufficient to acquire images from real environment. However, to capture images from a larger inspection site, a special acquisition system is required.

7. Conclusion

This paper presents the combined CNN-SVM model to solve the crack detection problem in concrete structure. The model considered CNN as a deep feature extractor and utilized SVM as the output classifier. The obtained results found that the proposed approach can detect cracks with the accuracy of 90.76 %. This is a promising result, which can be applied to assess concrete structures for inspection. The results show that the combination of CNN and SVM are promising, although the efficiency of the combined model can be further improved by fine-tuning CNN, like for example the size of convolution filter and max-pooling layer as define in section 3.3.1.

Acknowledgements

The valuable support and resources for the research received from the Faculty of Engineering, Thammasat University are greatly acknowledged.

References

[1] Chaiyasarn K, "Damage detection and monitoring for tunnel inspection based on computer vision," [Ph.D Dissertation]. U.K.: Christ's College, University of Cambridge England; 2011.

[2] Koch C, K. Georgieva, V. Kasireddy, B. Akinci and P. Fieguth, "A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure," Advanced Engineering Informatics, 2015;29(2):196-210.

[3] Yu S. N, Jang J. H.and Han C. S. "Auto inspection system using a mobile robot for detecting concrete cracks in a tunnel," Automation in Construction.2007 16(3):255-261.

[4] Niu X. X, Ching Y. Suen "A novel hybrid CNN–SVM classifier for recognizing hand-written digits." Pattern Recognition.2012;45(4):1318–1325.

[5] Fujita Y, Mitani Y. and Hamamoto Y. "A method for crack detection on a concrete structure," in Proc. IEEE 18th Int. Conf. Pattern Recognition (ICPR), 2006;4:901-904.

[6] Tong X, Guo J, Ling Y. and Yiu Z. "A new image-based method for concrete bridge bottom crack detection," in Proc. IEEE Int. Conf. Image Analysis and Signal Processing (IASP), 2011;568-571.

[7] Yamaguchi T and Hashimoto S. "Fast crack detection method for large size concrete surface images using percolation-based image processing," Machine Vision and Applications.2010;21(5):797-809.

[8] Yamaguchi T, and Hashimoto S. "Image processing based on percolation model." IEICE transactions on information and systems, 2006;89(7):2044-2052.

[9] Amhaz R, Chambon S, Idier J, and Baltazart V. "A new minimal path selection algorithm for automatic crack detection on pavement images." In Image Processing (ICIP), International Conference IEEE. 2014;17:788-792.

[10] Abdel-Qader I, Abudayyeh O. and Kelly Michael E, "Analysis of edge-detection techniques for crack identification in bridges," Journal of Computing in Civil Engineering. 2003;17(4):255-263.

[11] Abdel-Qader I, Pashaie-Rad S, Adudayyeh O. and Yehia S, "PCA-based algorithm for unsupervised bridge crack detection," Advances in Engineering Software. 2006;37(12):771-778.

[12] Cortes C, and Vapnik V. "Support-vector networks." Machine Learning, 1995;20(3):273-297.

[13] Liu Z, Shahrel A SUANDI, Takeshi OHASHI and Toshiaki EJIMA. "A Tunnel Crack Detection and Classification Systems Based on Image Processing," Machine vision application in industry inspection X. 2002;4664(145):820-8502.

[14] Moon H-G, and Kim Jung-H. "Intelligent crack detecting algorithm on the concrete crack image using neural network," Proceedings of the 28th ISARC.2011; 1461-1467.

[15] Chen Z, Derakhshani R. R, Halmen C, and Kevern J. T. "A texture-based method for classifying cracked concrete surfaces from digital images using neural networks." In Neural Networks (IJCNN), The International Joint Conference IEEE. 2011:2632-2637.

[16] Zou Q, Cao Y, Qingquan L, Qingzhou M, and Wang S. "CrackTree: Automatic crack detection from pavement images." Pattern Recognition Letters. 2012;33(3):227-238.

[17] Prasanna P, Dana K. J, Gucunski N, Basily B.B, Hung M. La, Ronny S. Lim, and H. Parvardeh. "Automated crack detection on concrete bridges," IEEE Transactions on Automation Science and Engineering, 2016;13(2);591-599.

[18] Shi Y, Cui L, Qi Z, Meng F. and Chen Z, "Automatic road crack detection using random structured forests." IEEE Transaction Intelligent Transportation Systems. 2016;17(12):1-12.

[19]Krizhevsky A, Sutskever I, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks," Advances in neural information processing systems 2012;25:1097-1105.

[20] LeCun Y, Bengio Y, and G. Hinton. "Deep learning. " Nature, 2015; 521(7553): 436-444.

[21] Luo G, An R, Wang K, Dong S, Zhang H "A deep learning network for right ventricle segmentation in short-axis MRI," In Computing in Cardiology Conference (CinC) IEEE, 2016;485-488.

[22] Shin H-C, Roth H. R, Gao M, Lu L, Xu Z, Nogues I, Yao J, Mollura D, and Summers R.M. "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning," IEEE transactions on medical imaging.2016;35(5):1285-1298.

[23] Tang Y. "Deep learning using linear support vector machines." In Workshop on Representational Learning, ICML, 2013;1306-0239.

[24] Ukai M, "Development of image processing technique for detection of tunnel wall deformation using continuously scanned image," Quart. Rep.–RTRI 2000;41(3):120–126.

[25] Xie D, Zhang L, and Bai L. "Deep Learning in Visual Computing and Signal Processing," Applied Computational Intelligence and Soft Computing. 2017;1-13.

[26] Makantasis K, Karantzalos K, Doulamis A, Doulamis N. Deep supervised learning for hyperspectral data classification through convolutional neural networks. In Geoscience and Remote Sensing Symposium (IGARSS), IEEE International 2015;4959-4962.

[27] Zhang L, Yang F, Daniel Y. Z, and Zhu Y. J, "Road crack detection using deep convolutional neural network," in Proceedings of the IEEE Int. Conf. on Image Processing (ICIP '16), 2016;3708–3712.

[28] Charles P.W.D, Keras, (2013) GitHub repository [Internet]. [cited 2017 March]. Available from:https://github.com/charlespwd/project-title

[29] Srivastava N, Hinton G, Krizhevsky A, Sutskever I, and Salakhutdinov R, Dropout: A simple way to prevent neural networks from overfitting, Journal Machine Learning Research"..2014;15(1):1929–1958.

[30] Pedregosa F, Varoquaux G, Gramfort A, V. Michel, B. Thirion, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J. "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, 2011;2825-2830.