A Deep Learning-based Multi-model Ensemble Method for Crack Detection in Concrete Structures

Luqman Ali[a], Farag Sallabi[a], Wasif Khan[a], Fady Alnajjar[a,*] and Hamad Aljassmi[b,c]

[a]Department of Computer Science and Software Engineering, College of Information Technology, UAEU, Al Ain 15551, United Arab Emirates
[b]Department of Civil Engineering, College of Engineering, UAEU, Al Ain 15551, United Arab Emirates
[c]Emirates Center for Mobility Research, UAEU, Al Ain 15551, United Arab Emirates
E-mail: 201990024@uaeu.ac.ae, f.sallabi@uaeu.ac.ae, 201990025@uaeu.ac.ae, fady.alnajjar@uaeu.ac.ae, h.aljasmi@uaeu.ac.ae

Abstract –

In civil infrastructures such as buildings, bridges, and tunnels, cracks are initial signs of degradation, which affect the structure's current and future performance adversely. Optimum maintenance plans in terms of cost and safety are important to evaluate the degree of deterioration of a structure. Manual inspection is usually performed, and cracks detected during inspections could help the inspectors to understand the damaged state of the concrete structures. However, these inspections are costly, laborious, and easily prone to human error. An automatic and fast crack detection at the earliest stage is crucial to avoid further degradation of the structure. In the past decades, various deep learning techniques have been introduced by researchers to automate the crack detection task. This paper introduces a deep learning-based multi-model ensemble approach for crack detection in concrete structures. The proposed architecture consists of five different customized convolutional neural networks (CNN) trained on data set created from two public datasets. The dataset consists of 8400 crack and non-crack images having a resolution of 224 * 224. Detailed experiments show that the majority voting ensemble technique shows better performance for crack detection in concrete structures. The accuracy of the individual CNN models 1, 2, 3, and 4 is recorded to be 95%,96%, 95%, and 97%, respectively, while the accuracy of the ensemble techniques is recorded to be 98%.

Keywords –

Crack Detection, Computer Vision, Automatic inspection, Convolutional Neural Networks, Ensemble Modeling, Concrete Cracks.

1 Introduction

Cracks are one of the first signs of degradation in any civil infrastructure, which requires proper attention and inspection in a timely manner. Traditionally, a team of experts carries out a visual inspection to check if there are any defects (cracking, spalling, defective joints, corrosion, potholes, etc.) in structures and report them for proper maintenance. Visual inspection is challenging and expensive, time-consuming, and laborious as it requires several trained professionals for the inspection. Moreover, visual inspection is not always reliable; failure to detect problems at the earliest stage can lead to disastrous effects. To overcome these limitations, a computer vision-based system is alternatively used to aid civil engineers during the inspection of concrete structures.

Normally, a vision-based crack detection system takes images as an input and gives them to the crack detection algorithm for classifying and localizing the cracks. The input images can be acquired using a digital camera, Unmanned Ariel Vehicle (UAV), or a smartphone. Most of the early crack detection systems are based on image processing methods such as thresholding [1], edge detection [2], filtering [3-6], fuzzy [7,8], percolation [9], and region growing [10]. However, the accuracy of these approaches heavily relies on the image focal length, quality, and capturing environment. Machine learning-based crack detection methods have improved the weaknesses of the rule-based approaches and can be categorized into traditional and deep learning approaches.

In traditional approaches, various features such as mean and variance [11], histogram [12,13], texture [14], Local Binary Patterns (LBP) [15] and multi-features [16] are extracted from the images. The obtained features are then given to various classifiers such as Support Vector Machine (SVM) [17,18], Decision Tree (DT), Genetic Algorithm [20], and various other classifiers for evaluation purposes. Feature extraction is a challenging task as it requires domain knowledge to extract the relevant information from the images which reflect the actual cracks.

To overcome the limitation of traditional approaches, an automatic feature learning technique such as deep learning is used, which automatically extracts useful features from raw images. Deep learning models have shown enormous performance in solving various concrete crack detection problems [21-23][33]. Zhang et al. [24] proposed CNN architecture consisting of six layers for crack detection in pavement structures using one million image patches of size 99×99. The patches are obtained from 500 images having a resolution of 3264×2448 collected by using a smartphone. Wang et al. [22] trained a five-layered CNN architecture for crack detection in pavement structure using 760 K image patches. Ali et al. [23] combined CNN architecture with a sliding window approach for crack detection and localization in pavement structure using 4k patches created from images acquired by using an unmanned aerial vehicle (UAV). Similarly, Cha et al. [25] performed CNN-based crack detection and sliding window-based crack localization in concrete structure using 40k images. Xu et al. [26] used 6k images and 28 layered end-to-end CNN architecture for crack detection in bridge structures. Pauly [27] investigate the effect of network depth on the performance of the pavement crack detection model and showed that network generalization ability could be enhanced by increasing the network depth. Zhang et al. [28] proposed CrackNet, a five-layered CNN architecture capable of detecting cracks in 3D asphalt surfaces at the pixel level. Due to the high computational time of the network and difficulty in detecting thin cracks, the authors improved the architecture by proposing CrackNet II [29], which can detect hairline cracks and has low processing time.

Transfer learning models make CNN more applicable with less computational cost Transfer learning models make CNN more applicable without incurring high computational costs or necessitating knowledge of how CNNs work. Gopalakrishnan et al. [30] developed a pavement crack detection model using pre-trained VGG 16 architecture using a small amount of training data. The proposed transfer learning model outperformed previous CNN models in terms of reliability, speed, and ease of implementation. Zhang and Chang [31] developed a pavement crack detection system using an Image-Net pre-trained model and 80k image patches. Wilson et al. [32] proposed a robust concrete crack detection model system based on the VGG-16 model using 3.5k images acquired by using UAV.

Although several deep learning algorithms are available for crack detection in concrete structures, however, none of them is completely accurate, and each method makes errors during prediction. It is still difficult to decide on the architecture and parameters of each model. Individual models may perform well for one classification task but not for another. Ensemble classifiers, on the other hand, aggregate the predictions of numerous independent models into a single prediction depending on criteria such as majority voting, unweighted and weighted average, and so on. Training multiple models and combining their predictions may result in better performance than individual models because the ensemble classifier explores a larger solution space from the set of individual classifier predictions. Therefore, this paper presents an ensemble classifier based on five different individual customized CNN models. The detailed experiments are evaluated on a dataset consisting of 8400 crack and non-crack images having a resolution of 224×224. The results are validated on different performance metrics such as accuracy, precision, recall, and F-1 score.

2 System Model and Assumptions

The proposed system is comprised of the three modules listed below:

(1) Database Creation
(2) Ensemble CNN modeling
(3) Classification Results.

In database creation, data is prepared and given as an input to CNN models built from scratch with varying parameters. The predictions obtained from the multi CNN models are combined by using ensemble methods such as Majority voting, weighted and unweighted average. Each module is discussed in detail below.

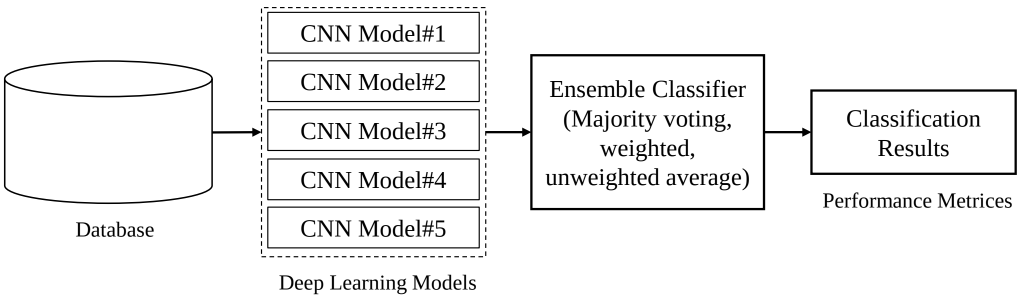

Figure 1: Overview of the proposed system

2.1 Database Creation

The dataset in the proposed system is created from publicly available datasets by Özgenel [34] and Dorafshan [35]. The main reason for combining the two datasets is to provide enough variance in the dataset's samples. The dataset consists of 16.8K image patches having a dimension of 224*224 pixels. As shown in Table 1, the patches were chosen at random from the datasets, with the split ratio for the training, validation, and testing sets being 60:20:20. Manual labeling was done for the crack and non-crack classes, each of which has an equal number of image patches.

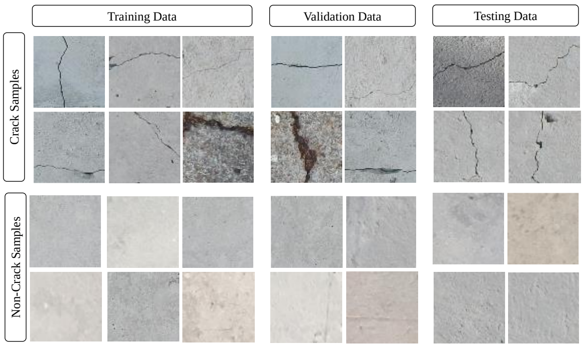

Figure 2: Samples of crack and non-crack patches in the dataset

Table 1: Total number of samples used in Training, validation, and Testing

Data	Training data		Validation Data Testing Data	
	Crack	Non-Crack	Crack	Non- Crack
16.8k	5.6k	5.6k	1.4k	1.4k

2.2 Convolutional Neural Network (CNN)

CNN is the most widely used deep learning network. CNN mainly comprises convolutional, activation, and pooling layers. The main function of these layers is to introduce non-linearity, extract relevant information (features) from input images and reduce its dimensionality to enhance network generalization. The function of each layer is described in detail below.

2.2.1 Convolutional Layer

CNN's convolutional layer extracts useful information from images and preserves the spatial relationship between its pixels. The filter slides over the image pixels, add them together and add bias to it to obtain the output feature vector as shown in Equation 1.

$$O = \sum(I_{k\times k} + W_{k\times k}) + Bias \qquad (1)$$

The convolution operation is performed on the input receptive field $I_{k\times k}$ where k represents the size of the kernel. $W_{k\times k}$ represent the filters weights which will be convolved over the input image, and B represent the filter bias. The obtained feature map is given as an input to the activation layer.

2.2.2 Max-Pooling Layer

The max-pooling layer performs a down sampling operation on the input array to reduce its dimensionality. The max-pooling layer divides the input array into small non-overlapping blocks and considers the maximum value of each block which helps in the reduction of model parameters and computational time.

2.2.3 Activation Layer

The activation layer performs an elementwise operation on the features coming from the convolutional layer to set the non-negative values to zero. This layer also introduces non-linearity to the feature map to ensure its usability. The mathematical operation of the activation layer is depicted in Equation 2 below.

$$\sigma(I) = \max(0, I) \qquad (2)$$

Where I represents the input feature vector.

2.2.4 Fully Connected Layer

The fully connected layer takes the results of the convolutional and max-pooling layer and performs logical inference on it. The input is flattening from 3D to 1D before giving as an input to the fully connected layer. The mathematical operation of a fully connected layer is shown in equation 3.

$$O_{V_o\times 1} = W_{V_o\times V_i}\, I_{V_i\times 1}\,.\, B_{V_o\times 1} \qquad (3)$$

Where O represents the output, V_i and V_o shows the size of the input and output vector. Additionally, the weight and matrix biased are represented by W and B.

2.2.5 Softmax Layer

The softmax layer is located at the end of the CNN architecture and is used for the prediction of classes. The softmax layer takes a vector of scores $x \in S^n$ and calculate probabilities $P \in S^n$ from the input scores. The mathematical operation is shown in Equation 4.

$$P = \begin{pmatrix} P_1 \\ \vdots \\ P_n \end{pmatrix} \text{ where } P = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \qquad (4)$$

Multiple models have been used in the proposed work. The summary of each model is shown in Table 2.

2.3 Ensemble Modelling

Ensemble CNN model consists of individually trained CNN models, which combines the prediction from multiple models to classify a new instance. In the proposed work, five different customized CNN models are used, and the predictions are aggregated to improve the system accuracy. Firstly, all the CNN models are trained individually on the same training data, and then the models are combined together for accurate prediction.

In the proposed work, three model different model combination techniques have been used: Majority voting, unweighted and weighted ensemble, as explained in detail below. The overall process of ensemble modeling is explained in Figure 3.

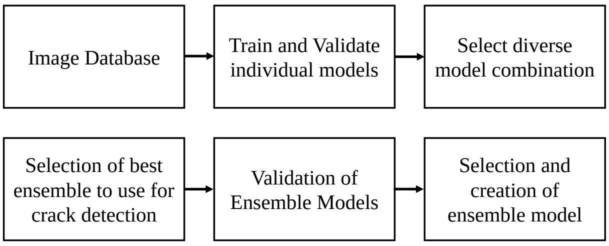

Figure 3: Representation of Ensemble Modeling process.

2.3.1 Majority Voting

In majority voting, the prediction of each model for a sample of data is known as a vote. The prediction having the majority votes from all the models is considered as the final prediction of the ensemble model. Suppose the prediction of CNN 1, 2, and 3 is label 0 while the prediction of CNN 4 and 5 is label 1. Then, the final prediction of the ensemble model is label 0 because of the majority votes. The mathematical representation of majority vote probabilities is shown in Equation (5) and (6) below.

$$\acute{P}_i = \frac{\sum_{j=1}^{n} M(P_{ij})}{n}, i = 1,2, \dots \dots \dots, m \quad (5)$$

Where $M(P_{ij}) = \begin{cases} 1 & if \;\; P_{ij} = \max(CNN(j)) \\ 0 & Otherwise \end{cases}$ (6)

Equation (2) combines all the votes assigned by the CNN model to j. n represent the total number of voters and are used for normalization purposes. In the majority voting approach, each input image will be assigned a class, and the vote of every model is considered equally without looking at their individual accuracies.

2.3.2 Unweighted and Weighted Ensemble

In an unweighted ensemble, the final prediction of the model is the average of the outcomes of all the CNN models. Averaging outcomes of CNN models decrease variance between them and increase the generalization ability of the ensemble model. The averaging of the CNN models' output is shown in Equation (7) in detail.

$$P_i^k = softmax^k(O_i) = \frac{O_i^k}{\sum_{j=1}^{n} \exp(O_j^k)} \quad (7)$$

In the above equation, n represent the number of classes, P_i^k is the output probability for unit i in class k, O_i^k is the output of kth CNN model for ith unit. In a weighted ensemble, weights are assigned to voters. The model is considered based on weighted majority voting and the sum of weighted probabilities. The weights to the voters are adjusted either by looking at their accuracies or by considering them as parameters and performing the optimal adjustment.

3 Experiments and Results

The experiments were performed on the dataset explained in section 2.1. The ensemble model consists of five different CNN models having different architecture and parameters, as shown in Table 2. CNN models and their ensembles are evaluated based on their accuracy, precision, recall, and F score as shown in Equation (8), (9), (10), and (11), respectively.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (8)$$

$$Precision = \frac{TP}{TP+FP} \quad (9)$$

$$Recall = \frac{TP}{TP+FN} \quad (10)$$

$$F_{1\,score} = 2 \times \frac{Precision\ .\ Recall}{Precision+Recall} \quad (11)$$

TP (True Positive) and TN (True Negative) values show the correctly identified crack and non-crack samples, while FP (False Positive) and FN (False Negative) represent the incorrectly identified crack and non-crack samples. All the experiments were performed using python programming on an Alienware Arura R8 core i9-9900k CPU @3.60 GHz desktop system with 32 GB RAM and an NVIDIA GeForce RTX 2080 GPU. The number of epochs for tall the models was chosen 20 as the loss of all the models reach a minimum level, and there was no further improvement in the model's accuracy. In the proposed work, five different CNN architectures were trained for 20 epochs which results in 100 trained networks. The best-performing trained network of each model was selected based on the evaluation metrics, as shown in Table 3.

In the proposed work, all the CNN architectures were built from scratch, and their various parameters were fine-tuned to achieve high performance. A comprehensive visual evaluation of all the CNN models was performed, the training and validation loss curves were plotted as shown in Figures 4,5, 6, 7 and 8. The confusion matrices of all the models are summarized in Table 3.

In CNN model 1, the number of parameters is 1.19 million, and the number of convolutional layers and max-pooling layers are 3 and 3, respectively. The accuracy

Table 2: Architecture and parameters of models

Model1	**Model2**	**Model 3**	**Model 4**	**Model 5**
Input Layer (224×224)				
Conv1	Conv1	Conv1	Conv1	Conv1
Actv1 (ReLU)	Max-Pool1	Actv1 (ReLU)	Actv1 (ReLU)	Max-Pool1
Max-Pool1	Actv1 (ReLU)	Max-Pool1	Max-Pool1	Actv1 (ReLU)
Dropout (0.05)	Conv2	Dropout (0.05)	Dropout (0.05)	Conv2
Conv2	Actv2 (ReLU)	Conv2	Conv2	Max-Pool2
Actv2 (ReLU)	Max-Pool2	Actv2 (ReLU)	Actv2 (ReLU)	Dropout (0.05)
Max-Pool2	Dropout (0.05)	Max-Pool2	Max-Pool2	Conv3
Dropout (0.05)	Conv3	Dropout (0.05)	Dropout (0.05)	Actv3 (ReLU)
Conv3	Actv3 (ReLU)	Conv3	Conv3	Max-Pool3
Actv3 (ReLU)	Conv4	Actv3 (ReLU)	Actv3 (ReLU)	Dropout (0.05)
Max-Pool3	Actv4 (ReLU)	Max-Pooling 3	Max-Pool3	Flatten1→FC1 →Actv5 →FC2
Dropout (0.05)	Max-Pool3	Dropout (0.05)	Dropout (0.05)	Softmax
Flatten1→FC1→Actv4 →FC2→Actv5	Dropout (0.05)	Conv4	Conv4	Parameters = 0.83
Softmax	Conv5	Actv4 (ReLU)	Actv4 (ReLU)	
Parameters= 1.19 M	Max-Pool4	Max-Pooling 4	Max-Pool4	
*conv = Convolutional. *Max-Pool= Max-pooling * FS = Filter Size *ReLU = Rectified Linear Unit *FC = Fully Connected *Actv : Activation layer *M = Millions	Conv6	Dropout (0.05)	Dropout (0.05)	
	Actv5 (ReLU)	Flatten1→FC1→Actv5→FC2	Conv5	
	Max-Pool5	Softmax	Actv5 (ReLU)	
	Dropout (0.05)	Parameters = 0.32	Max-Pool5	
	Flatten1→FC1→Actv6→FC2		Dropout (0.05)	
	Softmax		Flatten1→FC1→Actv6→FC2	
	Parameters = 0.92		Softmax	
			Parameters = 0.11	

- All convolutional Layers: (32, 3×3 convolutions, Stride= 1×1, No padding).
- All Max-pooling layer: Filter size (FS 3×3)

and loss graph of the training and validation of CNN model 1 is shown in Figure 4. In the graph, both the training and validation curves show little divergence, which shows that the model is not subjected to overfitting. The testing accuracy, precision, recall, and F1 score of CNN model is 0.928. 0.982, 0.921, and 0.951, respectively, as shown in Table 3.

Similarly, CNN model 2 consists of 6 convolutional and 5 model max-pooling layer. The architecture has a total of 0.92 million papers. The accuracy and loss graph shows that the architecture has less tendency towards overfitting, as shown in Figure 5. The testing accuracy, precision, recall, and F1 score of the CNN model 2 is 0.970, 0.994, 0.971, and 0.983, respectively.

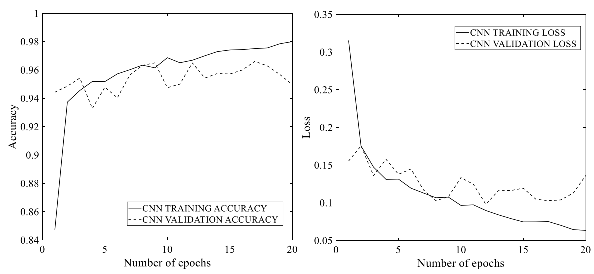

Figure 4: Training and Validation of CNN model 1 (Accuracy and loss graphs)

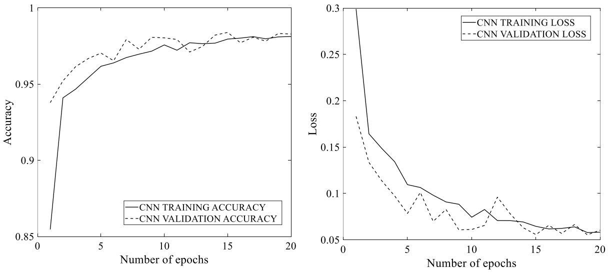

Figure 5: Training and Validation of CNN model 2 (Accuracy and loss graphs)

Figure 6 shows the accuracy and loss graph of training and validation of CNN model 3. The graph shows no signs of the model overfitting. The architecture of model 3 consists of 4 convolutional, 4 max-pooling, and 0.32 million parameters. The testing accuracy, precision, recall, and F1 score of the model is 0.953, 0.996, 0.950, and 0.973 respectively. Moreover, the number of parameters in CNN model 4 is 0.11 million. The architecture consists of 5 convolutional and 5 max-pooling layers.

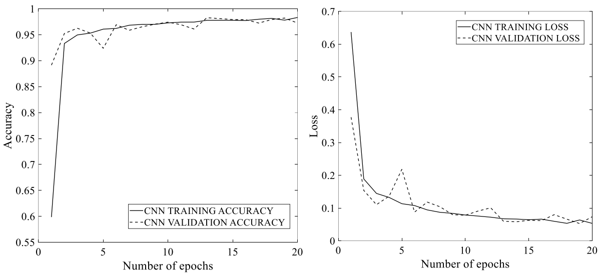

Figure 6: Training and Validation of CNN model 3 (Accuracy and loss graphs)

The accuracy and loss graph of CNN model 4 shows no overfitting, as shown in Figure 7. The testing accuracy, precision-recall and F score of the model are 0.966, 0.995, 0.967, and 0.981, respectively.

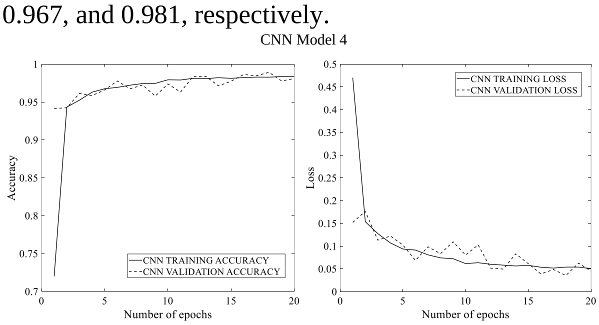

Figure 7: Training and Validation of CNN model 4 (Accuracy and loss graphs)

Moreover, the architecture of the CNN model 5 consists of 3 convolutional and 3 max-pooling layers and is having 0.83 million parameters. The accuracy and loss graph of the training and validation of the model is shown in Figure 8. The testing accuracy, precision, recall, and F score of the model is 0.976, 0.981, 0.982, and 0.974, respectively.

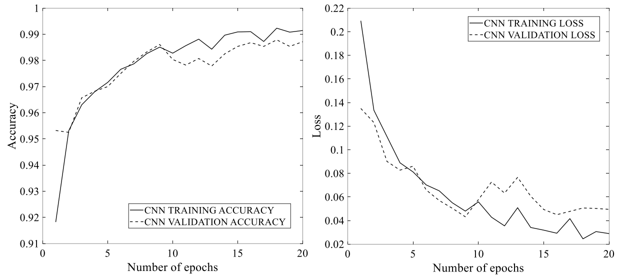

Figure 8: Training and Validation of CNN model 5 (Accuracy and loss graphs)

It can be seen from Table 3 that CNN model 2 and 4 outperform all the individual models in term of accuracy, precision, recall, and F1 score. The combined ROC curve of all the models is shown in Figure 6 below. To improve the accuracy further, ensemble modeling is used. In ensemble modeling, all the ensemble models i.e. the majority voting, unweighted average, and weighted average ensemble classifiers, achieved better results than individual models. The majority voting ensemble classifier achieved the highest testing accuracy of 0.991 with precision, recall, and F1 score of 0.996, 0.985, and 0.990, respectively. The testing accuracy, precision, recall, and F1 score of unweighted average ensemble classifiers are recorded 0.989, 0.995, 0.982, and 0.989, respectively. The testing accuracy of the weighted ensemble classifier is 0.990, which is slightly higher than the unweighted ensemble classifiers. Also, the value of precision-recall and F1 score of the weighted ensemble classifier is 0.997, 0.982, and 0.989, respectively.

4 Discussion

In the prosed work, a multi-model ensemble classifier is presented. The ensemble model combines the prediction of various customized CNN models by using various ensemble techniques such as the majority voting, unweighted average, and weighted average. The dataset is made from two publicly available datasets and contains 16.8k crack and non-crack patches. It can be seen in Table 3 that the ensemble models show better performance as compared to individual classifiers for crack and non-crack classification. The proposed models successfully achieved above 98% to classify between crack and Non-crack patches. It is found that all the proposed ensemble models achieve the best accuracy, precision, recall, and F1 score in comparison with the individual CNN models. The performance of individual CNN models (CNN Model 1, 2, 3, 4, and 5) are comparable with each other.

Table 3: Overall Experimental Results

Model	Class	Crk (0)	N-Crk (1)	Testing Accuracy	Precision	Recall	F score
Model1	Crk (0)	1369	25	0.928	0.982	0.921	0.951
	N-Crk (1)	116	1290				
Model 2	Crk (0)	1386	8	0.970	0.994	0.971	0.983
	N-Crk (1)	40	1366				
Model 3	Crk(0)	1389	5	0.953	0.996	0.950	0.973
	N-Crk (1)	72	1334				
Model 4	Crk (0)	1387	7	0.966	0.995	0.967	0.981
	N-Crk (1)	46	1360				
Model 5	Crk (0)	1378	26	0.976	0.981	0.968	0.974
	N-Crk (1)	45	1351				
E1: Majority Voting	Crk (0)	1389	5	0.991	0.996	0.985	0.990
	N-Crk (1)	31	1375				
E2: UnWeighted Average	Crk (0)	1387	4	0.989	0.995	0.982	0.989
	N-Crk (1)	42	1364				
E3: Weighted Average	Crk (0)	1390	4	0.990	0.997	0.982	0.989
	N-Crk (1)	28	1381				

Crk: Crack N-Crk: Non-crack

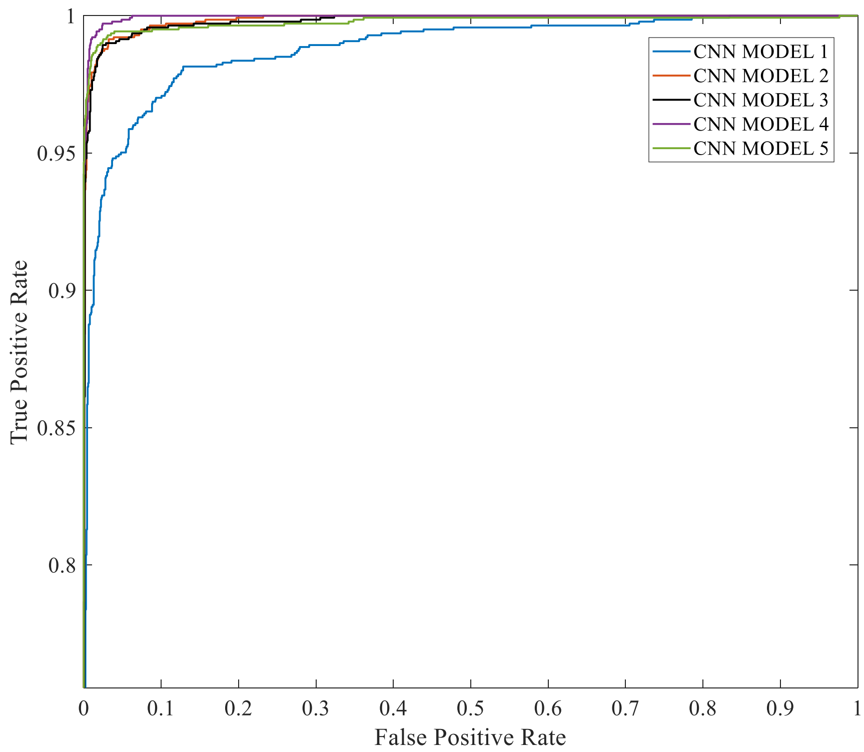

Figure 7: ROC curve (receiver operating characteristic curve) of all CNN individual models

The proposed model has the capability to enhance the performance of individual deep learning models and is very useful in the automatic detection of concrete cracks. The proposed framework is designed from a combination of less computational CNN architectures. The proposed system can be modified by adding more damage types of concrete structures in the dataset. The proposed framework can be easily used for a real-time robotic video inspection system. One of the drawbacks of the proposed system is its dependance on the base CNN models. If the accuracy of one of the models is degraded the overall accuracy will be affected.

5 Conclusion

In the proposed work, a new deep learning-based ensemble classifier is proposed by combining the predictions of various CNN models. The performance of the proposed Ensemble classifier is compared with individual CNN architectures in terms of testing accuracy, precision, recall, and F1 score. For the dataset creation, two publicly available datasets are selected and combined to provide variance between data samples. Extensive experiments were conducted by training individual CNN models to investigate their performance. The prediction from these models are then combine or ensembled to improve the performance of Concrete Crack Detection Model. From the experiments and above discussion, it can be concluded that the proposed multi-model ensemble classifier can be used for crack detection in concrete structures. As the current study was based on crack detection in static images, therefore, in the future, we are planning to explore crack detection in video streams of concrete structures using end to end deep learning techniques.

6 References

[1] Kamaliardakani, M.; Sun, L.; Ardakani, M.K. Sealed-Crack Detection Algorithm Using Heuristic Thresholding Approach. J. Comput. Civ. Eng. 2016, 30, 04014110.

[2] Abdel-Qader, I.; Abudayyeh, O.; Kelly, M.E. Analysis of edge-detection techniques for crack identification in bridges. J. Comput. Civ. Eng. 2003, 17, 255–263.

[3] Sinha, S.K.; Fieguth, P.W. Morphological segmentation, and classification of underground pipe images. Mach. Vis. Appl. 2006, 17, 21–31.

[4] Sinha, S.K.; Fieguth, P.W. Automated detection of cracks in buried concrete pipe images. Autom. Constr. 2006, 15, 58–72.

[5] Chambon, S.; Subirats, P.; Dumoulin, J. Introduction of a wavelet transform based on 2D matched filter in a Markov random field for fine structure extraction: Application on road crack detection. In Image Processing: Machine Vision Applications II; International Society for Optics and Photonics: Bellingham, WA, USA, 2009; Volume 7251, p. 72510A.

[6] Fujita, Y.; Hamamoto, Y. A robust automatic crack detection method from noisy concrete surfaces. Mach. Vis. Appl. 2011, 22, 245–254.

[7] Kim, K.B.; Cho, J.H. Detection of Concrete Surface Cracks using Fuzzy Techniques. J. Korean Inst. Inf. Commun. Eng. 2010, 14, 1353–1358.

[8] Choudhary, G.K.; Dey, S. Crack detection in concrete surfaces using image processing, fuzzy logic, and neural networks. In Proceedings of the 2012 IEEE Fifth International Conference on Advanced Computational Intelligence (ICACI), Nanjing, China, 18–20 October 2012; pp. 404–411.

[9] Yamaguchi, T.; Nakamura, S.; Saegusa, R.; Hashimoto, S. Image-Based Crack Detection for Real Concrete Surfaces. IEEE Trans. Electr. Electron. Eng. 2008, 3, 128–135.

[10] Li, Q.; Zou, Q.; Zhang, D.; Mao, Q. FoSA: F* Seed-growing Approach for crack-line detection from pavement images. Image Vis. Comput. 2011, 29, 861–872.

[11] Oliveira H., Correia P.L. Automatic road crack detection and characterization. *IEEE Trans. Intell. Transp. Syst.* 2013;**14**:155–168. doi: 10.1109/TITS.2012.2208630.

[12] Bray J., Verma B., Li X., He W. A neural network based technique for automatic classification of road cracks; Proceedings of the International Joint Conference on Neural Networks; Vancouver, BC, Canada. 16–21 July 2006; pp. 907–912.

[13] Xu Z., Zhao X., Song H., Tao L., Na W. Asphalt pavement crack recognition algorithm based on histogram estimation and shape analysis. Chin. J. Sci. Instrum. 2010;31:2260–2266.

[14] Lu Z.W., Wu C.D., Chen D.Y., Shang S.B. Pavement crack detection algorithm based on sub-region and multi-scale analysis. *J. Northeast. Univ.* 2014;**35**:622–625.

[15] Chen F.C., Jahanshahi M.R., Wu R.T., Joffe C. A texture based video processing methodology using bayesian data fusion for autonomous crack detection on metallic surfaces. Comput.-Aided Civ. Inf. Eng. 2017;32:271–287. doi: 10.1111/mice.12256.

[16] L. Ali, S. Harous, N. Zaki, W. Khan, F. Alnajjar and H. A. Jassmi, "Performance Evaluation of different Algorithms for Crack Detection in Concrete Structures," 2021 2nd International Conference on Computation, Automation and Knowledge Management (ICCAKM), 2021, pp. 53-58, doi: 10.1109/ICCAKM50778.2021.9357717.

[17] Gavilán, M.; Balcones, D.; Marcos, O.; Llorca, D.F.; Sotelo, M.A.; Parra, I.; Ocaña, M.; Aliseda, P.; Yarza, P.; Amírola, A. Adaptive Road Crack Detection System by Pavement Classification. Sensors 2011, 11, 9628–9657.
[18] Sari, Y.; Prakoso, P.B.; Baskara, A.R. Road Crack Detection using Support Vector Machine (SVM) and OTSU Algorithm. In Proceedings of the 2019 6th International Conference on Electric Vehicular Technology (ICEVT), Bali, Indonesia, 18–21 November 2019; pp. 349–354.
[19] Sheng, P.; Chen, L.; Tian, J. Learning-based road crack detection using gradient boost decision tree. In Proceedings of the 2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA), Wuhan, China, 31 May–2 June 2018; pp. 1228–1232.
[20] Feulvarch, E.; Fontaine, M.; Bergheau, J.M. XFEM investigation of a crack path in residual stresses resulting from quenching. Finite Elem. Anal. Des. 2013, 75, 62–70.
[21] Fei, Y.; Wang, K.C.; Zhang, A.; Chen, C.; Li, J.Q.; Liu, Y.; Yang, G.; Li, B. Pixel-Level Cracking Detection on 3D Asphalt Pavement Images Through Deep-Learning- Based CrackNet-V. IEEE Trans. Intell. Transp. Syst. 2020, 21, 273–284.
[22] Wang, K.C.P.; Zhang, A.; Li, J.Q.; Fei, Y.; Chen, C.; Li, B. Deep Learning for Asphalt Pavement Cracking Recognition Using Convolutional Neural Network. In Airfield and Highway Pavements 2017: Design, Construction, Evaluation, and Management of Pavements, Proceedings of the International Conference on Highway Pavements and Airfield Technology, ASCE American Society of Civil Engineers 2017, Philadelphia, PA, USA, 27–30 August 2017; American Society of Civil Engineers: Reston, VA, USA, 2017; pp. 166–177.
[23] Ali, L., Valappil, N. K., Kareem, D. N. A., John, M. J., & Al Jassmi, H. (2019, November). Pavement Crack Detection and Localization using Convolutional Neural Networks (CNNs). In 2019 International Conference on Digitization (ICD) (pp. 217-221). IEEE.
[24] Zhang, L.; Yang, F.; Zhang, Y.D.; Zhu, Y.J. Road crack detection using deep convolutional neural network. In Proceedings of the International Conference on Image Processing, ICIP, Phoenix, AZ, USA, 25–28 September 2016; pp. 3708–3712.
[25] Cha, Y.J.; Choi, W.; Büyüköztürk, O. Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks. Comput. Civ. Infrastruct. Eng. 2017, 32, 361–378.
[26] Xu, H.; Su, X.; Wang, Y.; Cai, H.; Cui, K.; Chen, X. Automatic Bridge Crack Detection Using a Convolutional Neural Network. Appl. Sci. 2019, 9, 2867
[27] Pauly, L.; Hogg, D.; Fuentes, R.; Peel, H. Deeper networks for pavement crack detection. In Proceedings of the 34th ISARC, Taipei, Taiwan, 28 June–1 July 2017; pp. 479–485.
[28] Zhang, A.; Wang, K.C.; Li, B.; Yang, E.; Dai, X.; Peng, Y.; Fei, Y.; Liu, Y.; Li, J.Q.; Chen, C. Automated Pixel-Level Pavement Crack Detection on 3D Asphalt Surfaces Using a Deep-Learning Network. Comput. Aided Civil Infrastruct. Eng. 2017, 32, 805–819.
[29] Zhang, A.; Wang, K.C.; Fei, Y.; Liu, Y.; Tao, S.; Chen, C.; Li, J.Q.; Li, B. Deep Learning–Based Fully Automated Pavement Crack Detection on 3D Asphalt Surfaces with an Improved CrackNet. J. Comput. Civ. Eng. 2018, 32, 04018041.
[30] Gopalakrishnan, K.; Khaitan, S.K.; Choudhary, A.; Agrawal, A. Deep Convolutional Neural Networks with transfer learning for computer vision-based data-driven pavement distress detection. Constr. Build. Mater. 2017, 157, 322–330.
[31] Zhang, K.; Cheng, H.D.; Zhang, B. Unified Approach to Pavement Crack and Sealed Crack Detection Using Preclassification Based on Transfer Learning. J. Comput. Civ. Eng. 2018, 32, 04018001.
[32] da Silva, W.R.L.; de Lucena, D.S. Concrete Cracks Detection Based on Deep Learning Image Classification. Proceedings 2018, 2, 489.
[33] Ali L, Alnajjar F, Jassmi HA, Gocho M, Khan W, Serhani MA. Performance Evaluation of Deep CNN-Based Crack Detection and Localization Techniques for Concrete Structures. Sensors. 2021; 21(5):1688. https://doi.org/10.3390/s21051688.
[34] Dorafshan, S., Thomas, R. J., & Maguire, M. (2018). SDNET2018: An annotated image dataset for non-contact concrete crack detection using deep convolutional neural networks. Data in brief, 21, 1664-1668.
[35] Özgenel, Ç. F., & Sorguç, A. G. (2018). Performance comparison of pretrained convolutional neural networks on crack detection in buildings. In ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction (Vol. 35, pp. 1-8). IAARC Publications.