Deep learning-enabled vision-based pavement crack detection

Eshta Ranyal[1], Ayan Sadhu[2], Kamal Jain[1]
[1]Department of Civil Engineering, IIT Roorkee, India.
[2]Department of Civil and Environmental Engineering, Western University, London, ON, Canada
Email : eshtaranyal@iitr.ac.in, asadhu@uwo.ca.

ABSTRACT: Cracks in pavements are common distress affecting the safety and longevity of transportation infrastructure. Automated pavement monitoring for cracks is a pliable solution with access to high-performance computing and memory resources. Vision-based deep learning (DL) techniques promise economic and automated results using commercially available cameras. To maximize the advantage of DL methods, this article proposes a vision-based method employing an optimized and regularized *RetinaNet* convolutional neural network (CNN) for detecting concrete cracks in pavements. The designed CNN is trained on a subset of the SDNET2018 dataset with an F1 score of 94%. The evaluation metrics of the model promise a realistic solution for detecting cracks in concrete pavements.

KEYWORDS: CNN, concrete cracks, pavement distress, pavement monitoring, RetinaNet.

1 INTRODUCTION

Pavement infrastructure endures deterioration owing to age, traffic load, varying weather conditions and inadequate maintenance. Timely reinforcement of pavements through regular monitoring to prevent structural losses can ensure a well-functioning transportation sector, a healthy ecosystem, and a flourishing economy. Pavement preventative maintenance necessitates corrective measures before a catastrophic deterioration occurs. Conventionally, manual and semi-automated approaches are used to assess the condition of deteriorating pavements by skilled engineers. However, these approaches are labor-intensive, time-consuming, limited to human bias, expertise, terrain, and weather conditions, making them unreliable. Vision-based pavement surface assessment integrated with deep learning (DL) [1] offers immense potential in simplifying and reducing these limitations and the cost dimension of maintenance methods.

With the advancement of DL, significant research has been carried out in pavement distress monitoring [2]. The majority of studies focus on developing DL models to determine the presence or absence of distress and its type. Of the various distress types, crack detection and analysis have been at the center of research as cracks are among the most recurring distress types. Cha et al. (2017) [3] used about 40,000 images of damaged and undamaged concrete for detecting cracks in concrete structures using CNN. Zhang et al. (2017) [4] proposed a pixel-level CNN to detect cracks on 3D pavement surfaces. The proposed CNN, CrackNet, was more efficient than traditional CNNs because of invariant image size through all layers for pixel-perfect accuracy. Tong et al. (2017) [5] proposed an automated pavement crack-length detection algorithm using a five-layer-deep CNN for crack lengths between 0-8 cm. In addition, the *k*-means clustering analysis was used to extract the length and shape of each pavement crack accurately. Another pavement crack detection approach was investigated by Gopalakrishnan et al. (2017-2018) [6,7] using transfer learning-based deep CNN to compare the performance of various classifiers for crack detection. Fan et al. (2018) [8] proposed CNN based on images acquired from an iPhone to detect pavement cracks. The proposed methodology had accuracy superior to traditional machine learning (ML) techniques. Similarly, Maeda et al. (2018) [9] used two object detection methods, Single-Shot Multibox Detector (SSD) using Inception V2 and MobileNet, to detect road surface damage from images acquired using a dashboard-mounted smartphone to classify the pavement cracks into various damage classes. In another study, Zhang et al. (2018) [10] proposed an algorithm to classify sealed and unsealed cracks in asphalt pavement using a transfer learning-based deep CNN.

Similarly, J. Li et al. (2019) [11] performed pavement distress detection using Faster-RCNN on a dataset. The data-driven artificial intelligence-based methodologies are favorable in terms of computational speed and efficiency and promise effective solutions in terms of pavement crack detection. In another approach, Roberts et al. (2020) [12] employed prevailing DL methods to analyze municipality road networks, emphasizing pavement distress types and related severities. Opara et al. (2021) [13] classified pavement cracks into longitudinal, transverse, alligator cracks, and potholes using the deep learning model YOLO v3 for localization and detection of cracks.

The DL techniques have led to a series of breakthroughs in vision-based classification. Deep networks integrate multi-level (low/mid/high) features and classifiers in an end-to-end multilayer fashion. However, classification-based approaches fail to localize the damage, thus limiting the system to class-based identification. Recent years have seen a growth in exploring DL methodologies for localizing cracks. This paper presents a crack classification and localization approach by using a single-stage deep learning CNN algorithm to perform classification and bounding box regression to locate the pavement damage in the form of cracks.

2 METHODOLGY

RetinaNet, a single-stage CNN optimized to best detect concrete cracks in pavements, is proposed in this paper.

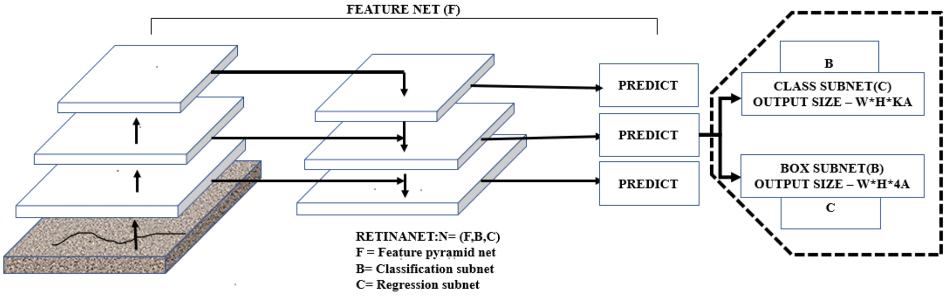


Figure 1. *RetinaNet* network architecture.

A single-stage detector directly acts over a dense sampling of probable locations. In contrast, two-stage detectors first propose a region of interest, and the classifier then processes only these regions of interest. Thus the advantage of using *RetinaNet*, a single-stage detector, is in its computational speed and simplicity compared to two-stage detectors. The overall detection procedure is described in Fig 1. In this process, the pavement image acts as an input to the *RetinaNet* network. The *ResNet50* backbone framework calculates multi-scale feature maps, which are laterally connected to the feature pyramid net that allows to up-sample the feature maps and merge the corresponding layers with the same spatial size. The classification and regression subnet predict the probability of the presence of potholes and regresses the offset for bounding boxes, respectively.

2.1 Databank Generation

SDNET2018 [14], a concrete crack image dataset for ML applications, is used in this study. This dataset is an amalgamation of crack and non-crack images acquired using a commercially available 16 MP Nikon digital camera. A subset of 1000 pavement images with 256x256 pixel resolution, is extracted from this dataset. It is ensured that pavement images of varying textures and varying light conditions with shadow effects are selected for an all-encompassing and unbiased database. The images are annotated for cracks using the *LabelImg* graphical annotation tool. During the annotation process, the labels and bounding boxes for 1046 instances in 1000 images are assigned. The training, validation, and testing datasets are randomly selected from the annotated images and split into three disjoint sets. The training dataset comprised approximately 60% of the images, while the validation and test dataset contained 20% each.

2.2 Implementation

All experiments were performed using the open-source *RetinaNet* library [15], Python3.8.11, CUDA 10.2, and CUDNN 7.0 on a workstation with Core i7-8700k @3.2 GHz CPU, 32 GB DDR4 memory, and 16 GB NVIDIA GeForce GTX 1070 graphics processing unit (GPU). The *RetinaNet* network was trained with Adam optimizer set to a learning rate of 1e-5 for 27,600 iterations, in batches of 4, for a total of 200 epochs. Optimization by hyper-parameter tuning via trial-and-error was performed, and a series of training rounds led to the optimal combination of parameters.

3 RESULTS AND DISCUSSION

The *RetinaNet* model is trained for 200 epochs in batches of 4 for a total duration of 10hrs. To ensure an optimized model, several rounds of training are carried out on the dataset. Initially, the model is trained with default parameters to assess its performance. To increase the model's performance, hyperparameter tuning is performed. In addition, dropout regularization is implemented by randomly dropping neurons in the backbone framework to prevent the model from overfitting the training dataset. The best performance of the model is achieved after ensuring a well-balanced and uniformly distributed dataset. The efficiency of a trained model is likewise determined by its loss metrics which should minimize during training optimization and by monitoring the evaluation metrics in terms of recall, precision, and F1-score. During the training process, the classification and regression loss are monitored for each epoch, and it is observed that the loss decreases with each epoch. The classification, regression, and overall loss after each epoch for the training dataset are represented in Figure 2.

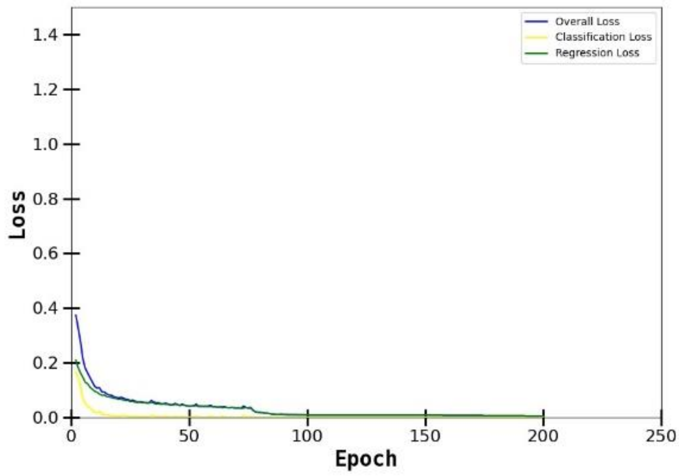


Figure 2. Loss vs Epoch

Table 1. Overall classification loss metric.

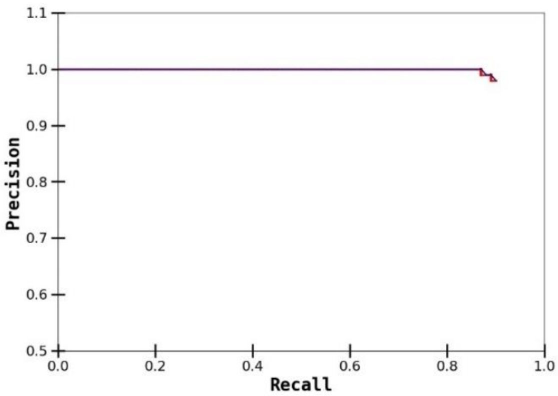
Loss	@epoch 1	@epoch 250
Classification	0.682	0.005
Regression	0.268	0.00
Overall	0.951	0.005

Table 1 represents the initial and final values of the losses during training. The performance of the trained model was evaluated on the train, validation, and test dataset by calculating the precision, recall, and F1-score for each of the datasets, as illustrated in table 2. Recall, precision, and F1 score are chosen as the evaluation metrics and have a strong dependence on the total number of true and false positives, which are determined with respect to the ground truth annotation.

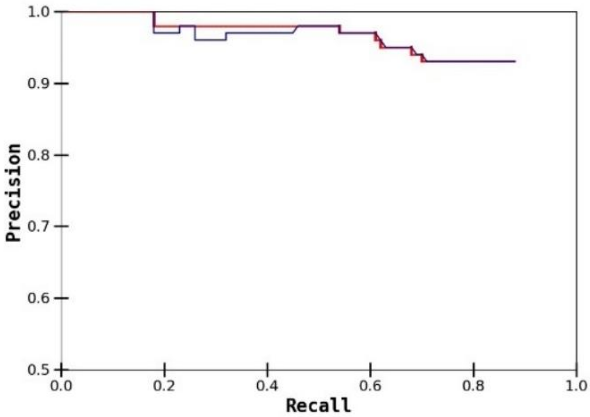
Table 2. Evaluation metrics of the hypertuned model.

Precision			Recall			F1-score		
Train	Val	Test	Train	Val	Test	Train	Val	Test
0.98	0.93	0.95	0.90	0.88	0.93	0.94	.90	0.94

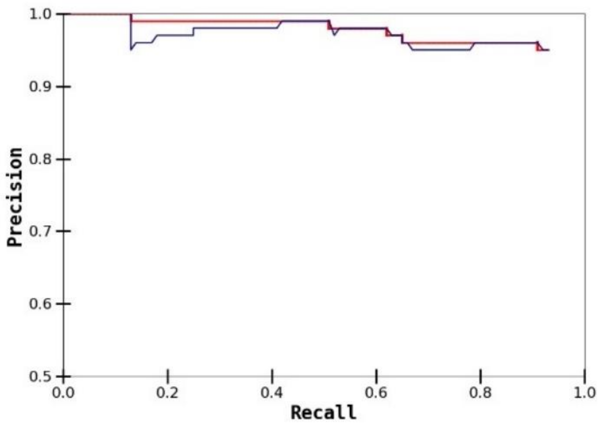
A schematic representation of the variation of precision and recall measures for each of the datasets are plotted in Figure 3 for a better interpretation of the model's ability to detect cracks. Evaluation of the model on the test dataset revealed a reasonable performance with a precision of 95%, recall of 93%, and an F1 score of 94%.



(a)



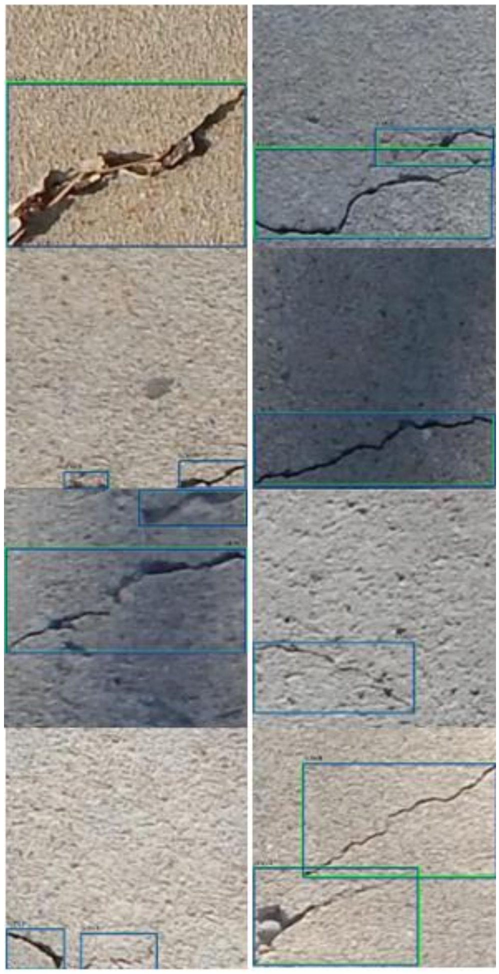
(b)



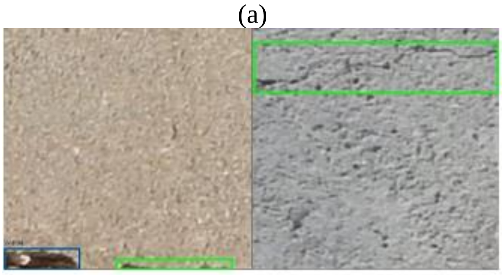
(c)

Figure 3. Precision vs recall (a) training dataset (b) validation dataset (c) testing dataset.

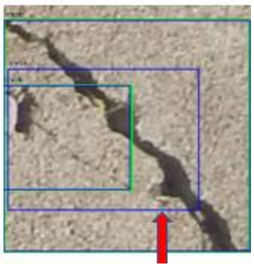
The model is evaluated on the test dataset to provide a clear understanding of how the model performs on images unknown to the trained model.



(a)



(b)



(c)

Figure 4. Samples of detection on the test dataset (a) True positives (b) False negatives (c) False positives.

An algorithm is implemented to visualize the results of the test images. Sample results from the proposed method are illustrated in Figure 4. The green-colored boxes represent the ground truth values, whereas the blue boxes are the detections made by the model. It can be seen that the model successfully detects cracks of varying shapes, sizes, and depths. It is imperative to note that despite the varying textures and shadow effects of the concrete pavements, as is seen in Figure 4 (a), the model successfully detects the cracks. Figure 4 (a) represents the true positive instances that have been detected by the model with high accuracy. However, there are some scenarios where the model fails to predict the cracks and account for false negatives. The false-negative detections by the trained model are shown in Figure 4(b). The false-positive detection is illustrated in Figure 4(c), which accounts for cracks detected by the model which are not present.

3.1 Computational time and complexity

The computational complexity of an algorithm is defined by its time and memory requirements. The proposed model employs ResNet50 as its backbone framework and thus successfully extracts the required discriminative features of pavement cracks with 50 convolutional layers. The computational speed of any CNN model is inherently dependent on its backbone architecture, and increasing its layers has a direct effect on the model’s complexity and its execution time to process each image. With a backbone framework of 50 layers detection speed of our model is approximately .07s on average to process an image.

4 CONCLUSIONS

This paper proposes a vision-based cost-effective approach for detecting cracks in concrete pavements using a single-stage CNN architecture, *RetinaNet*. An optimized and regularized DL algorithm is implemented, which successfully detects cracks in concrete pavements. The following conclusions are drawn from this study:

1) Commercially available cameras are effective in pavement crack inspections.
2) The proposed model performs exceptionally well with a high F1-score of 94% on unseen data, thus promising usability in automating the pavement distress detection process.

The usability of a single-stage CNN over two-stage CNNs is validated by the model's computational time of 0.07 seconds per image on average. The model’s high computational speed is also attributed to the model’s backbone framework of 50 layers which is just apt in terms of depth. The database SDNET2018 employed in this study is a combination of concrete crack images taken under varying light conditions and with different texture complexities. The proposed model is not limited to data bias, as is justified by the model's performance on an impartial and random distribution of 1k images in training, validation, and test datasets. From the results, it is observed that the model was especially strong at detecting cracks of small size and could easily distinguish neighboring cracks. The model successfully distinguishes between pavements with and without cracks, while at the same time localizing the crack positions. Thus, it can be concluded that the proposed model is well suited for the classification and localization of cracks in concrete pavements and ensures the proposed model's applicability in real-time application.

Despite the proposed model's high performance, there are certain limitations. A common limitation observed in vision-based CNN approaches is the failure to determine the cracks' internal characteristics due to the 2D nature of images. In the future, methods to assess the crack severity will be explored to extract meaningful information that can be integrated with the proposed model to determine the crack severity by measuring crack width or depth.

REFERENCES

[1] S. Sony, K. Dunphy, A. Sadhu, and M. Capretz, “A systematic review of convolutional neural network-based structural condition assessment techniques,” *Engineering Structures*, vol. 226. Elsevier Ltd, Jan. 01, 2021. doi: 10.1016/j.engstruct.2020.111347.

[2] W. Cao, Q. Liu, and Z. He, “Review of Pavement Defect Detection Methods,” *IEEE Access*, vol. 8, pp. 14531–14544, 2020, doi: 10.1109/aCCESS.2020.2966881.

[3] Y. J. Cha, W. Choi, and O. Büyüköztürk, “Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 5, pp. 361–378, May 2017, doi: 10.1111/mice.12263.

[4] D. Zhang, Q. Li, Y. Chen, M. Cao, L. He, and B. Zhang, “An efficient and reliable coarse-to-fine approach for asphalt pavement crack detection,” *Image and Vision Computing*, vol. 57, pp. 130–146, Jan. 2017, doi: 10.1016/j.imavis.2016.11.018.

[5] Z. Tong, J. Gao, and H. Zhang, “Recognition, location, measurement, and 3D reconstruction of concealed cracks using convolutional neural networks,” *Construction and Building Materials*, vol. 146, pp. 775–787, Aug. 2017, doi: 10.1016/j.conbuildmat.2017.04.097.

[6] K. Gopalakrishnan, H. Gholami, A. Vidyadharan, and A. Agrawal, “Crack Damage Detection in Unmanned Aerial Vehicle Images of Civil Infrastructure Using Pre-Trained Deep Learning Model,” *International Journal for Traffic and Transport Engineering*, vol. 8, no. 1, pp. 1–14, 2018, doi: 10.7708/ijtte.2018.8(1).01.

[7] K. Gopalakrishnan, S. K. Khaitan, A. Choudhary, and A. Agrawal, “Deep Convolutional Neural Networks with transfer learning for computer vision-based data-driven pavement distress detection,” *Construction and Building Materials*, vol. 157, pp. 322–330, Dec. 2017, doi: 10.1016/j.conbuildmat.2017.09.110.

[8] Z. Fan, Y. Wu, J. Lu, and W. Li, “Automatic Pavement Crack Detection Based on Structured Prediction with the Convolutional Neural Network,” Feb. 2018, [Online]. Available: http://arxiv.org/abs/1802.02208

[9] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama, and H. Omata, “Road Damage Detection and Classification Using Deep Neural Networks with Smartphone Images,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 12, pp. 1127–1141, Dec. 2018, doi: 10.1111/mice.12387.

[10] K. Zhang, H. D. Cheng, and B. Zhang, “Unified Approach to Pavement Crack and Sealed Crack Detection Using Preclassification Based on Transfer Learning,” *Journal of Computing in Civil Engineering*, vol. 32, no. 2, p. 04018001, Mar. 2018, doi: 10.1061/(asce)cp.1943-5487.0000736.

[11] J. Li, X. Zhao, and H. Li, “Method for detecting road pavement damage based on deep learning,” Apr. 2019, p. 81. doi: 10.1117/12.2514437.

[12] R. Roberts, G. Giancontieri, L. Inzerillo, and G. di Mino, “Towards low-cost pavement condition health monitoring and analysis using deep learning,” *Applied Sciences (Switzerland)*, vol. 10, no. 1, Jan. 2020, doi: 10.3390/app10010319.

[13] J. N. Opara, A. B. B. Thein, S. Izumi, H. Yasuhara, and P. J. Chun, “DEFECT DETECTION ON ASPHALT PAVEMENT BY DEEP LEARNING,” *International Journal of GEOMATE*, vol. 21, no. 83, pp. 87–94, Jul. 2021, doi: 10.21660/2021.83.6153.

[14] S. Dorafshan, R. J. Thomas, and M. Maguire, “SDNET2018: An annotated image dataset for non-contact concrete crack detection using deep convolutional neural networks”, doi: 10.15142/T3TD19.

[15] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal Loss for Dense Object Detection.”