A Data-Driven Framework for Structural Crack Identification in 3D Mobile LiDAR Scans Using Deep Learning Classification Models

Yasmine Megahed [1], Omar Elsisi [2], Ahmed Shaker [3]

[1] Department of Civil Engineering, Toronto Metropolitan University, 350 Victoria Street
Toronto, ON M5B 2K3, Canada - yasmine.megahed@torontomu.ca
[2] Department of Civil Engineering, Toronto Metropolitan University, 350 Victoria Street
Toronto, ON M5B 2K3, Canada - oelsisi@torontomu.ca
[3] Department of Civil Engineering, Toronto Metropolitan University, 350 Victoria Street
Toronto, ON M5B 2K3, Canada - ahmed.shaker@torontomu.ca

Keywords: Mobile LiDAR, Deep Learning, Pavement Crack Detection, Point Cloud Classification, Multi-Layer Perceptron (MLP), Radiometric Indices.

Abstract

In cold-climate regions like Canada, pavement infrastructure deteriorates rapidly due to extreme freeze-thaw cycles and heavy use of de-icing salts, accelerating the formation of structural cracks and imposing a financial burden on municipal budgets. By providing an automated LiDAR (Light Detection and Ranging)-based detection framework, this research offers a cost-effective, high-precision monitoring tool that enables early intervention, reducing long-term repair costs and enhancing road safety across Canadian provincial networks. This study evaluates the performance of Support Vector Machines (SVMs) and Multi-Layer Perceptrons (MLPs) for crack classification in a multi-dimensional feature space. We propose integrating geometric height (H) with a novel set of radiometric indices, including the Normalized Difference Intensity Index (NDII) and the Green Ratio (GR), to enhance classification stability. Results demonstrate that both SVM and MLP achieved comparable accuracies of 87% and 86%, respectively, in low-dimensional feature spaces. A critical analysis of the MLP learning curves reveals that the introduction of NDII acted as a numerical stabilizer, mitigating the oscillations caused by raw brightness fluctuations. Furthermore, the study identifies an information ceiling, as architectural expansion of the MLP improved convergence stability but did not exceed the 87% accuracy threshold. These findings provide a robust framework for automated road maintenance using stabilized radiometric features in LiDAR-based distress identification.

1. Introduction

Common asphalt pavement cracks encompass fatigue (alligator) cracking, longitudinal and transverse cracking, block cracking, and edge cracking. These distress types arise from different mechanisms such as repeated traffic loading, thermal contraction, binder aging, and inadequate support. For instance, fatigue cracking indicates structural wear from traffic, whereas transverse and block cracks result from thermal contraction and binder aging. Accurate crack detection is critical to determine a suitable intervention, as maintenance strategies can range from sealing and localized repairs to overlays and full-depth rehabilitation (Miller and Billinger, 2014; ASTM International, 2018).

Pavement crack detection is crucial for maintaining road infrastructure, as cracks often precede potholes and more severe structural failures. Preventive treatments are substantially less expensive than resurfacing or reconstruction, but they depend on timely, reliable detection at scale. Hence, early detection enables timely, cost-effective repairs, enhancing public safety (Nguyen *et al.*, 2023). This challenge is particularly pronounced in Canada, where the climate is a primary driver of pavement deterioration due to extreme weather conditions. As temperatures frequently fall below 0°C, Frequent freeze-thaw cycles and cold-weather stresses cause asphalt to crack more rapidly than in milder climates, accelerating pavement degradation (Maadani *et al.*, 2021; ClimateData, 2026). In regions like Ontario and Quebec, water infiltrates minor cracks during winter, freezes, and expands, leading to larger surface failures (Yang *et al.*, 2020). Consequently, automated crack detection systems are increasingly vital for efficient, large-scale management of these issues (Fahad *et al.*, 2022).

Canadian municipal and provincial pavement preservation programs reflect this reality. For example, the City of Barrie describes crack sealing as an early-life preservation activity typically performed within the first 5–7 years, when transverse cracking begins due to frost expansion (City of Barrie, 2026). These precautionary measures result from the realization of the tangible cost gap between early and late interventions. Public reporting in Ontario notes that minor cracks limited to the top layer can be sealed for about $7,500/km of cracks. At the same time, resurfacing costs are on the order of $180,000 lane-km, underscoring why early sealing is economically attractive when it delays restoration (Office of the Auditor General of Ontario, 2016). Bid-based unit price reporting in Alberta also provides practical magnitudes, including crack routing and sealing at $14.47/m and pressure-injected epoxy crack repair at $23.71/m (Alberta Transportation, 2016).

Public sector response is implemented primarily through provincial and municipal asset management practices, supported by standards and program guidance. Canadian best-practice guidance on sealing and filling cracks is published through the National Guide to Sustainable Municipal Infrastructure (Federation of Canadian Municipalities, 2003). In Ontario, pothole prevention and repair program guidance explicitly includes rout-and-seal among eligible preventive strategies and links implementation to standardized specifications (Government of Ontario, 2025). For bridge structures, Ontario regulations require biennial inspections and reference standardized procedures in the Ontario Structure Inspection Manual (Government of Ontario, 2010; Office of the Auditor General of Ontario, 2021). These constraints motivate the development of automated detection systems that are accurate, auditable, and scalable across heterogeneous networks.

Crack detection techniques have evolved from traditional Image Processing methods to feature-engineered Machine Learning and, more recently, Deep Learning Networks that learn features end-to-end, and 3D/LiDAR (Light Detection and Ranging) approaches that incorporate radiometric and geometric evidence. Early Image Processing methods relied on handcrafted operations (*i.e.,* Canny edge detection) and deterministic pipelines that combined contrast enhancement, edge/ridge detection, thresholding, morphological refinement, and wavelet-based filters to identify cracks. While these were computationally efficient, straightforward, and did not require labelled data, they faltered in real-world scenarios involving shadows, noise, and varied pavement textures, as they were highly sensitive to parameter choices and could fail when shadows, stains, joints, and aggregate texture resembled cracks. Reviews consistently highlight that the main difficulty is robustness under domain shift (lighting, texture, sensor changes, seasonality), and that performance reporting is often not directly comparable across tasks and metrics (Mohan and Poobal, 2018; Cao *et al.,* 2020; Hsieh and Tsai, 2020; Munawar *et al.,* 2021; Kheradmandi and Mehranfar, 2022; Fahad *et al.*, 2022; Nguyen *et al.*, 2023; and Zhang *et al.*, 2025).

Subsequent Machine Learning methods used extracted features (*i.e.*, intensity, gradient/texture, wavelets, structured tokens, edges, histograms), which were then classified using Support Vector Machines (SVMs) or Random Forests (Nguyen *et al.*, 2023). These approaches improved accuracy but still required extensive feature engineering and struggled with generalization across diverse environments (Fahad *et al.*, 2022). CrackTree is a representative milestone that combined shadow handling and tensor voting to infer a crack probability map and enforced global connectivity through graph construction (Zou *et al.*, 2012). Reproducibility also improved through public implementations such as CrackIT (Oliveira and Correia, 2014). CrackForest then adopted random structured forests to learn crack-like tokens and detect cracks more robustly than purely hand-tuned pipelines (Shi *et al.*, 2016). Despite these gains, such approaches still depend heavily on feature design and re-tuning across acquisition settings (Cao *et al.*, 2020; Hsieh and Tsai, 2020).

Deep Learning has become dominant by learning representations directly from data rather than relying on handcrafted features. Early Convolutional Neural Network (CNN)-based crack detectors showed strong performance using patch-based training and helped establish supervised learning as a practical baseline (Zhang *et al.*, 2016). CNNs automatically learn hierarchical features from raw data, enabling end-to-end detection. Architectures such as U-Net, YOLO, Faster R-CNN, and DeepLab have demonstrated robust performance (Zhang *et al.*, 2025). Variants of U-Net often produce precise crack masks, while YOLO models support near-real-time detection (Saberironaghi and Ren, 2024). Overall, Deep Learning offers superior accuracy and robustness, especially for pixel-level crack segmentation (Nguyen *et al.*, 2023). Soon after, encoder–decoder segmentation architectures reframed crack detection as a dense prediction problem, improving localization and enabling measurement-oriented outputs (Munawar *et al.*, 2021; Kheradmandi and Mehranfar, 2022). CNNs were also shown to generalize beyond pavements to concrete crack imagery under challenging conditions (Cha *et al.*, 2017). However, reviews repeatedly report persistent limitations in cross-region generalization and inconsistent evaluation protocols, which can inflate apparent improvements when task definitions differ (Hsieh and Tsai, 2020; Munawar *et al.*, 2021).

Appearance-based detectors remain vulnerable to illumination and shadowing. 3D sensing adds geometric evidence that can reduce these confounds and supports georeferenced reporting over large networks. CrackNet-style models demonstrated pixel-level crack detection on 3D measurements of asphalt surfaces and motivated further improvements to 3D learning pipelines (Zhang *et al.*, 2017; Zhang *et al.*, 2018). At the same time, 3D crack detection introduces practical challenges: point density varies with range and speed, fine cracks can approach the noise floor, and high-quality labels are costly and ambiguous, which motivates simple, reproducible baselines that can be extended toward richer 3D descriptors and multi-modal fusion (Cao *et al.*, 2020; Hsieh and Tsai, 2020).

Deep Learning achieves the best performance on curated datasets, while 3D sensing provides additional invariances and better compatibility with network-level asset management. The major remaining gaps are the standardized cross-dataset evaluation with explicit domain-shift reporting, label-efficient learning and adaptation to reduce annotation effort, and operational quality assurance (uncertainty and failure reporting) needed for deployment (Hsieh and Tsai, 2020; Munawar *et al.*, 2021; Kheradmandi and Mehranfar, 2022).

Research in crack detection draws from diverse datasets. Standard ones like Crack500 and DeepCrack feature Red, Green, and Blue (RGB) images with corresponding crack masks. Advanced datasets incorporate thermal imagery, depth maps, LiDAR point clouds, and multimodal data (Zhang *et al.*, 2025). Some are synthetic, generated via simulations or Generative Adversarial Networks (GANs) to augment variety. Recent trends favour drone- and vehicle-mounted acquisition. Datasets such as 3DCrack, derived from high-resolution laser scanning, provide detailed 3D information across varying conditions. However, there remains a need for larger, more diverse real-world datasets to bolster model generalization (Sabouri and Sepidbar, 2023; Zhang *et al.*, 2025).

Given the high cost of fully annotated crack datasets, emerging paradigms focus on reducing labelling requirements and enhancing generalization (Zhang *et al.*, 2025):

Semi-supervised learning: Integrates limited labelled data with unlabelled images, often employing contrastive learning (Zhang *et al.*, 2025).
Weakly-supervised learning: Relies on coarse annotations, such as image-level labels. Vision-language models, such as Contrastive Language-Image Pretraining (CLIP), have been adapted for prompt-based segmentation (*i.e.,* CrackCLIP) (Liang *et al.*, 2025).
Unsupervised/anomaly detection: Learns patterns of regular pavement and detects deviations, commonly via GAN reconstruction (Nguyen *et al.*, 2023).
Few-shot learning: Enables detection with minimal examples through meta-learning (Zhang *et al.*, 2025).
Domain adaptation: Enables detection with minimal examples through meta-learning (Zhang *et al.*, 2025).
Foundation models: Large, pre-trained models such as CLIP and Segment Anything Model (SAM) are fine-tuned for crack segmentation. Combinations such as SAM with Detectron2 achieve high mean intersection over union (mIoU) scores (Liang *et al.*, 2025).

These paradigms, particularly those involving foundation and transformer-based models, substantially boost zero- and few-shot performance across datasets (Zhang *et al.*, 2025).

https://doi.org/10.5194/isprs-archives-XLIX-B1-2026-185-2026

Different studies have addressed the crack detection problem from various angles:

Method evolution: Transitioning from Image Processing to Machine Learning and Deep Learning, with hybrid models (*i.e.,* CNN features combined with SVM) is still under exploration (Nguyen *et al.*, 2023; Zhang *et al.*, 2025).
Task perspective: Crack detection can be framed as classification, object detection, or semantic segmentation. Segmentation models like U-Net or DeepLab yield the most detailed outputs (Fahad *et al.*, 2022).
Pipeline perspective: Many studies address the entire workflow from data acquisition (using drones, survey vehicles, or laser scanners) to preprocessing and integration with asset management systems, emphasizing challenges like variable lighting and practical deployment (Zhang *et al.*, 2025).

Classification in crack detection occurs at multiple levels (Gupta and Dixit, 2022): **Image/patch-level** determines presence (crack vs. no-crack), **Object-level** identifies and labels specific crack instances, and **Pixel-level** that assigns type or severity labels per pixel.

Contemporary pipelines often integrate segmentation for localization with classification for type or severity assessment. CNN-based models such as DeepCrack and U-Net variants are prevalent for segmentation (Zhang *et al.*, 2016; Yang *et al.*, 2019). In the absence of pixel masks, patch- or image-level supervision suffices (Shi *et al.*, 2016). Recent advancements have extended YOLO, Faster R-CNN (Ren *et al.*, 2015; Redmon *et al.*, 2016), and transformers for multitask detection and severity estimation (Guo *et al.,* 2023). Despite datasets like Crack500 and CFD, challenges persist, including class imbalance, narrow cracks, low contrast, and cross-environment generalization (Yang *et al.*, 2019; Shi *et al.*, 2016).

To address the limitations of traditional intensity-based crack detection, especially for radiometrically manipulated or poor-weather-condition point clouds, this research aims to develop a robust, automated framework for extracting structural cracks from mobile LiDAR point clouds. The primary objective is to investigate the impact of an expanded feature space, integrating radiometric indices, on the classification accuracy. Furthermore, this study examines how these indices enhance the stability and convergence of the deep learning process by reducing feature overlap. By implementing a deep Multi-Layer Perceptron (MLP) architecture, the work evaluates the effectiveness of spectral-spatial fusion for reliable pavement assessment in complex urban environments. Figure 2 visualizes close-ups of some asphalt cracks from both datasets.

2. Study Area and Dataset

We analysed two datasets of mobile LiDAR data acquired in 2022 by the Multi-Purpose LiDAR System (MPLS) (Elshorbagy, 2020) for cracked asphalt at the parking lots at Oakville GO Station in the Greater Toronto Area (GTA), ON, Canada (Figure 1). The system used a single-intensity Velodyne VLP-16 sensor to collect data with an average point spacing of 17 cm at a 903 nm laser wavelength. Trimple POSPac and CloudCompare software initially processed the point clouds after acquisition before we used them in this study. Dataset-1 has 2,702,910 points, whereas Dataset-2 has 1,241,529 points and was coloured in a separate project using high-spatial-resolution RGB orthophotos from the Scholars GeoPortal.

Figure 1. Point Clouds’ Ground Projection on Google Earth Basemaps.

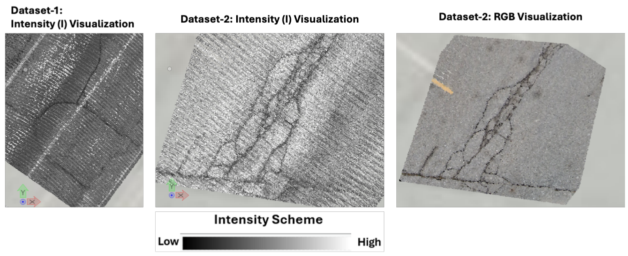

Figure 2. Cracks’ Close-ups.

3. Methods and Experimental Work

3.1 Overview

Figure 3 schematically explains the proposed methodology applied in this research. It is structured into four main phases: elevation normalization, feature space expansion through radiometric indices, sample data collection, and point-based classification.

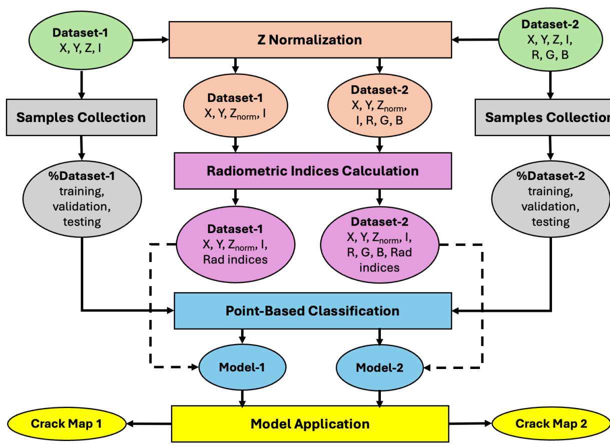

Figure 3. Methodology.

The proposed computational framework was developed using Python, leveraging a robust ecosystem of specialized libraries for geospatial and machine learning tasks. The Laspy library ensures efficient handling of LAS/LAZ file formats, whereas the NumPy and Matplotlib libraries were used for numerical operations and for visualizing the model's learning curve, respectively. The machine learning pipeline was implemented using the Scikit-learn library, and the deep MLP architecture was constructed and trained using the Keras Application Programming Interface (API) (Keras, 2015).

3.2 Elevation Normalization

To enhance the model's sensitivity to fine-scale surface variations, the Z-coordinate was normalized by subtracting the minimum elevation value from each point. This zero-level normalization was crucial to transform absolute spatial coordinates into relative height values, effectively aligning the geometric scale with the radiometric indices. By shifting the data to a common baseline, we prevented the gradient descent used in the subsequent classification step from being biased toward large-magnitude elevations. This process ensured that the classification models could effectively prioritize subtle vertical depressions, a characteristic of structural cracks, thereby achieving faster convergence and improving overall learning stability.

3.3 Feature Space Expansion

To overcome the spectral ambiguity and feature overlap inherent in raw LiDAR intensity data, we expanded the input feature space by integrating multispectral attributes and derived radiometric indices. While laser intensity is sensitive to surface roughness, distinguishing asphalt cracks from similar low-reflectivity features (*i.e.*, tire marks or damp pavement) remains challenging. Therefore, we calculated a set of specialized indices to enhance the contrast between structural distress and the surrounding road surface. Specifically, we integrated the Brightness (Br), Normalized Difference Intensity Index (NDII), and the Green-Red (GR) ratio to leverage the complementary nature of laser returns and optical RGB data.

The overall Br averages the RGB channels to provide a baseline for illumination intensity, NDII highlights the contrast between reflected red light and the backscattered laser intensity, and the GR ratio differentiates between organic debris and actual pavement distress. These indices, represented in Equations 1, 2, and 3, are inspired by (Kang *et al.*, 2016), (Yan *et al.*, 2015), and (Gamon *et al.*, 1992), respectively. We applied this action to Dataset-2 using its intensity and RGB characteristics, as Dataset-1's feature space is limited to normalized Z-values and laser intensity, with no room for further expansion.

$$Br = \frac{R + G + B}{3} \tag{1}$$

where Br = overall brightness
R, G, B = point's spectral values in the red, green, and blue wavelength ranges, respectively.

$$NDII = \frac{R - I}{R + I} \tag{2}$$

where $NDII$ = Normalized Difference Intensity Index
R = point's spectral values in the red wavelength range

$$GR = \frac{G}{R + \varepsilon} \tag{3}$$

where GR = Green-Red Ratio
G, R = point's spectral values in the green and red, wavelength ranges, respectively
ε = small constant to avoid division by zero

3.4 Sample Data Collection

We employed ERDAS IMAGINE for precise point-level sampling. We manually digitized polygon features representing two distinct classes: crack (structural distress) and non-crack (background asphalt and road markings). The resulting point cloud was subsequently split into three subsets: a testing set (20%) and a development set (80%). We applied a secondary hold-out split on the development set to ensure unbiased model evaluation and prevent overfitting. Specifically, 20% was reserved as a validation set for hyperparameter tuning and early stopping, while the remaining 80% (forming approximately 64% of the total points) was used for model training. The training/validation split is a random division performed during the modelling step, ensuring consistency across runs despite different splits. The train_test_split function was employed in Python for the hierarchical data partitioning. This hierarchical strategy, detailed in Figure 4, ensures that the final testing results reflect the model's performance on unseen pavement scenarios.

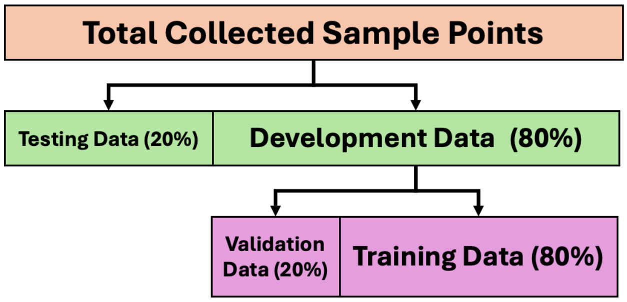

Figure 4. Hierarchical Sample Data Split Strategy.

In the initial stages of model development, data duplication (oversampling) was explored to mitigate the significant class imbalance between pavement cracks and the background asphalt. The compute_class_weight function in Python was used to address potential class imbalances between crack and non-crack samples.

3.5 Point-Based Classification

We used Dataset-1 in the initial phase of this study. We implemented the SVM classifier (Megahed *et al.*, 2021) as a baseline to evaluate the discriminative potential of the height (H) and intensity (I) attributes using the Scikit-learn Python module with standard hyperparameters: Radial Basis Function (RBF) kernel, penalty parameter C = 1.0, and the gamma coefficient was determined using the *scale* option (Pedregosa *et al.*, 2011).

We used Keras in a Python environment (Keras, 2015) to build 4 architectures of traditional feedforward deep MLP neural networks, as shown in Table 1. We designed 9 MLP models (Model-2 to Model-10) based on these architectures to simulate 6 feature-space-dependent Deep Learning classification scenarios, as described in Table 2. The dimensions of each input layer are equal to the number of features in the corresponding input feature space, and the number of neurons in each output layer was always set to the number of output classes: 2, crack and non-crack. The models were compiled using the Adam optimizer (Kingma and Ba, 2014) and the Categorical Cross-Entropy loss function (Goodfellow *et al.*, 2016). The Rectified Linear Unit (ReLU) function (Nair and Hinton, 2010) activated the input and hidden layers, whereas the Softmax function (Goodfellow *et al.*, 2016) activated the output layer. In all MLPs, we applied Sequential modelling and Dense layers.

We designed the classification scenarios to examine the effect of height (H) and intensity (I) on the crack detection accuracy. The RGB properties of Datase-2 enabled a thorough study of the gradual feature expansion by adding RGB, BR, NDII, and GR, as further demonstrated in Table 2.

Arc-w (Table 1) is the fourth MLP architecture we built to consider weighting imbalanced data samples, so they have nearly equal shares in the classification process, as mentioned in 3.4. It uses a Dropout layer as the first hidden layer with a rate of 0.1 to improve generalization by randomly dropping/deactivating 10% of neurons at each training iteration; this forces the model to learn more robust, redundant features (Srivastava *et al.*, 2014). The dropout mechanism is only active during training, while all neurons remain fully active during final classification and testing.

Arc	Attribute	Layers			# of Epochs	# of Batches
		Input	Hidden	Output		
Arc-w	# of layers	1	3	1	1000	500
	# of Neurons	16	Dropout (0.1), 12, 8	2		
Arc-1	# of layers	1	2	1	300	150
	# of Neurons	16	12, 8	2		
Arc-2	# of layers	1	2	1	700	300
	# of Neurons	16	12, 8	2		
Arc-3	# of layers	1	3	1	700	400
	# of Neurons	64	32, 16, 8	2		

Table 1. Specifications of Proposed MLP Architectures.

Dataset	Feature Space	Classifier	Scenario #	Model #
1	H, I	SVM	1	1
		MLP: Arc-1		2
2	H, I, RGB	MLP: Arc-w	3	3
	H, I	MLP: Arc-2	1	4
	H, RGB		2	5
	H, I, RGB		3	6
	H, I, RGB, Br		4	7
	H, I, RGB, Br, NDII		5	8
	H, I, RGB, Br, NDII, GR		6	9
	H, I, RGB, Br, NDII, GR	MLP: Arc-3		10

Table 2. Proposed Classification Scenarios.

We used standard Confusion Matrices to compute F1-scores to evaluate the performance of the 10 classification models. To ensure a rigorous assessment of the models' accuracies, the Overall Accuracy (OA) alone is insufficient, given the inherent class imbalance between pavement distress features and the background road surface. As shown in Equation 4, the F1-score is calculated as a function of Precision and Recall. These metrics represent the ratio of correctly predicted positive observations to the total predicted positives and the total actual positives, respectively (Megahed *et al.*, 2021).

$$F1-score = \frac{2 * Precision * Recall}{Precision + Recall} \quad (4)$$

where *F1-score* = harmonic mean of Precision and Recall of the crack class
Precision = ratio of correctly predicted positive observations to total predicted positives
Recall = ratio of correctly predicted positive observations to total actual positives

4. Results

Figure 5 illustrates the assessment results for the 10 classification models, represented as F1 Scores, and Figure 7 presents the deep MLP learning curves for the 9 MLP models. The major remarks are discussed below.

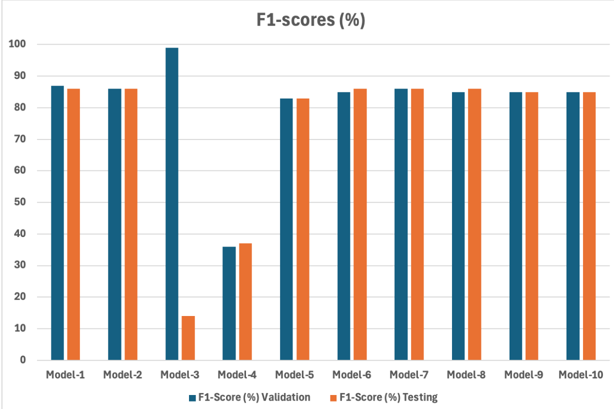

Figure 5. Classification Models' Accuracies.

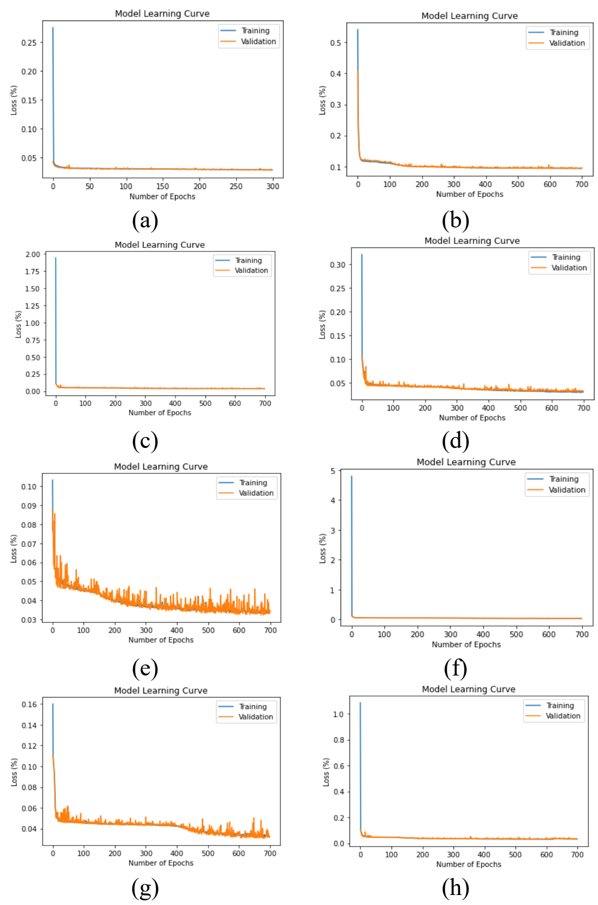

Figure 6. Deep MLPs' Learning Curves. (a) Model-2, (b) Model-4, (c) Model-5, (d) Model-6, (e) Model-7, (f) Model-8, (g) Model-9, (h) Model-10

In Dataset-1's height (H) and intensity (I) feature space, SVMs (Model-1) achieved 87% accuracy on the validation samples, very close to that of a deep MLP (Model-2) (86%). With fewer parameters to tune in a 2D space, whose values provided distinct physical signatures for pavement cracks, the SVMs maintained high generalization, and both models effectively extracted all available discriminative information. Consistent results from both models on the test data (86%) confirmed the models' robustness. Moreover, Model-2's learning curve (Figure 6.a) exhibits smooth convergence and a minimal gap between the training and validation losses. This stability indicates effective weight optimization and strong generalization across the pavement dataset. Figure 7 shows a close-up of the crack map of Dataset-1 classified by Model-1.

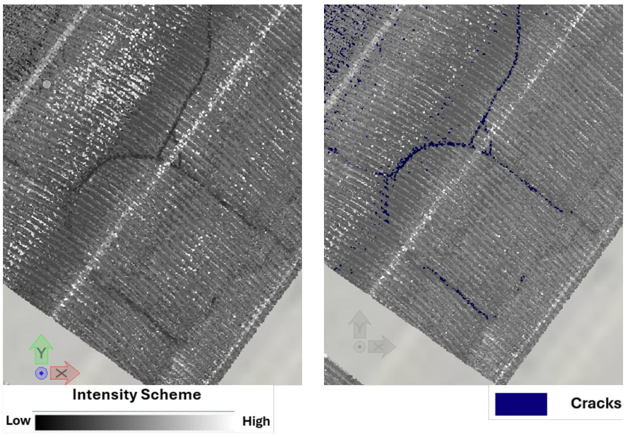

Figure 7. Dataset-1's Cracks Map Close-up. Left: Intensity Visualization of Row Data, Right: Cracks on top.

Model-4 examined the same feature space in Dataset-2; however, it achieved only 36% accuracy, suggesting that the dataset was acquired under adverse weather conditions, specifically during a rainy period, which increased moisture on the pavement surface and significantly influenced the laser backscatter intensity. Yet the model could achieve optimal generalization with stable convergence (Figure 6.b).

Model-5 tested the height values (H) along with the RGB attributes, rather than its non-discriminative intensity (I) field. This combination boosted the crack detection accuracy to 83% (Figure 5) and maintained a generalization success (Figure 6.c), concluding that RGB can compensate for intensity values when ruined by poor environmental conditions or data correction. Model-6 included the intensity (I) in the feature space of the previous scenario, increasing the accuracy to 85% (Figure 5) and still yielding robust learning (Figure 6.d).

Model-3 sought to further enhance this result by using weighted samples while retaining the same feature space: H, I, and RGB. Despite the 99% accuracy output on the validation samples, the model almost failed to detect cracks on the testing set (14%) (Figure 5), in a clear indication of overfitting due to data leakage. The weighting process caused the model to memorize specific instances rather than learning generalizable features. Consequently, the model achieved near-perfect scores on the biased validation set but failed to generalize to the independent testing set, which represents the true, natural distribution of pavement conditions.

Model-7 expanded the previous feature space to include the calculated radiometric indices, starting with the Br, which did not tangibly enhance crack detection accuracy (from 85% to 86%) (Figure 5), and increased generalization volatility (Figure 6.e). While the training loss converged, the Validation loss oscillated, highlighting the classifier's sensitivity to the additional input dimension.

Model-8 introduced NDII to Model-7's feature space. The index maintained the same crack-detection accuracy (85%; Figure 5) but neutralized the oscillations previously observed with the Br feature. Unlike raw intensity sums, the normalized ratio within NDII mitigates radiometric noise and compensates for variations in laser backscatter caused by surface geometry or sensor distance. This normalization provided a more consistent and discriminative signal for crack detection, leading to improved numerical stability in the MLP's gradient descent and a smoother learning curve (Figure 6.f).

Model-9 accumulated the last computed index, the GR ratio, in the feature space. This introduction maintained the same classification accuracy (85%; Figure 5); however, the validation curve shows minor late-stage oscillations, indicative of the increased dimensionality of the input space (Figure 6g). In response, Model-10 is a successful attempt to overcome these minor oscillations by expanding the MLP's architecture (Figure 6.h), while maintaining the same accuracy (85%; Figure 5), indicating that the model had already reached the information ceiling of the provided feature space. While the larger network optimized the mathematical mapping more quickly, it could not extract additional discriminative patterns because the inherent overlap between classes (cracks vs. non-cracks) in the physical data remained unchanged, suggesting the need to introduce different attributes (*i.e.*, geometric-based fields). Figure 8 shows a close-up of the crack map of Dataset-2 classified by Model-10.

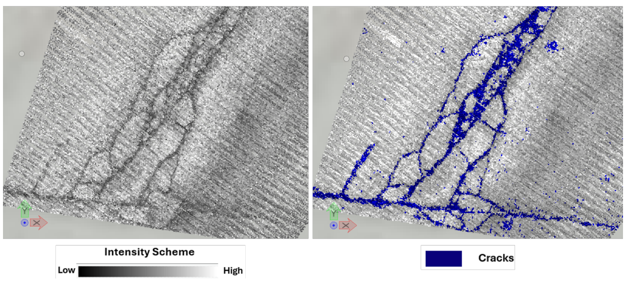

Figure 8. Dataset-2's Cracks Map Close-up. Left: Intensity Visualization of Row Data, Right: Cracks on top.

Conclusion

This research highlights the efficacy of integrating height and radiometric features from mobile LiDAR point clouds for automated pavement crack detection. The comparative analysis shows that SVM and MLP yield nearly identical performance when restricted to high-impact features such as height (H) and intensity (I), emphasizing the importance of feature quality over model complexity in low-dimensional spaces. The proposed NDII and GR indices significantly improved the signal-to-noise ratio, leading to smoother model convergence and enhanced generalization. The use of normalized ratios proved essential in neutralizing the adverse effects of varying environmental conditions (*i.e.*, surface moisture in Dataset-2) on laser backscatter. Architectural scaling of the MLP demonstrated that while deeper networks achieve faster and more stable numerical convergence, the final classification accuracy is primarily governed by the discriminative power of the input features. Future work will explore the deployment of geometric-based indices and evaluate their performance across diverse asphalt compositions and extreme weather scenarios.

References

Alberta Transportation, 2016: Unit Price Averages Reports. [Online]. Available: https://www.transportation.alberta.ca/content/doctype257/ production/unitpricelist.pdf

ASTM International, 2018: Standard Practice for Roads and Parking Lots Pavement Condition Index Surveys. [Online]. Available: https://www.astm.org/d6433-18.html

Cao, W., Liu, Q. and He, Z., 2020. Review Of Pavement Defect Detection Methods. *IEEE Access*, 8, pp.14531-14544.

Cha, Y.J., Choi, W. and Büyüköztürk, O., 2017. Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks. *Computer-Aided Civil and Infrastructure Engineering*, 32(5), pp.361-378.

City of Barrie, 2026: 2026 Pavement Management [Online]. Available: https://www.barrie.ca/planning-building-infrastructure/infrastructure/infrastructure-projects/pavement-management-program

ClimateData, 2026: Transportation Module Overview. [Online]. Available:https://climatedata.ca/transportation-module-overview/

Elshorbagy, A.M.A.; 2020: A Crosscutting Three-Modes-Of-Operation Unique LiDAR-Based 3D Mapping System Generic Framework Architecture, Uncertainty Predictive Model and SfM Augmentation. *Doctoral dissertation at Toronto Metropolitan University*.

Fahad, M., Nagy, R., Guangpin, L., and Rosta, S., 2022: Pavement Crack Monitoring: Literature Review. *Iraqi Journal of Civil Engineering*, 16(2), 76-89.

Federation of Canadian Municipalities, 2003: Guidelines for Sealing and Filling Cracks in Asphalt Concrete Pavement. [Online]. Available: https://fcm.ca/sites/default/files/documents/resources/guide/infraguide-guidelines-sealing-filling-cracks-asphalt-concrete-pavements-mamp.pdf

Gamon, J. A., Penuelas, J., & Field, C. B. (1992). A Narrow-Waveband Spectral Index That Tracks Diurnal Changes in Photosynthetic Efficiency. *Remote Sensing of Environment*, 41(1), 35-44.

Goodfellow, I., Bengio, Y., Courville, A. and Bengio, Y., 2016. Deep Learning (Vol. 1, No. 2, pp. 1-800). *Cambridge: MIT Press*.

Government of Ontario, 2010: O. Reg. 472/10: Standards for Bridges. [Online]. Available: https://www.ontario.ca/laws/regulation/r10472

Government of Ontario, 2025: Pothole Prevention and Repair Program. [Online]. Available: https://forms.mgcs.gov.on.ca/dataset/03b855c0-7928-4273-aae5-9a5be5383ae5/resource/8d826d82-eaf1-4ed6-9e2b-2a4bff822cd0/download/pothole-prevention-and-repair-program-guide-english.pdf

Guo, F., Qian, Y., Liu, J. and Yu, H., 2023: Pavement crack detection based on transformer network. *Automation in Construction*, 145, p.104646.

Gupta, P. and Dixit, M., 2022. Image-Based Crack Detection Approaches: A Comprehensive Survey. *Multimedia Tools and Applications*, 81(28), pp.40181-40229.

Hsieh, Y.A. and Tsai, Y.J., 2020. Machine Learning for Crack Detection: Review and Model Performance Comparison. *Journal of Computing in Civil Engineering*, 34(5), p.04020038.

Kang, Z., Yang, J. and Zhong, R., 2016. A Bayesian-Network-Based Classification Method Integrating Airborne Lidar Data with Optical Images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(4), pp.1651-1661.

Keras, 2015: A Superpower for ML Developers. [Online]. Available: https://github.com/fchollet/keras

Kheradmandi, N. and Mehranfar, V., 2022. A Critical Review and Comparative Study on Image Segmentation-Based Techniques for Pavement Crack Detection. *Construction and Building Materials*, 321, p.126162.

Kingma, D.P. and Ba, J., 2014. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv*:1412.6980.

Liang, F., Li, Q., Yu, H., and Wang, W., 2025; CrackCLIP: Adapting Vision-Language Models for Weakly Supervised Crack Segmentation. *Entropy*, 27(2), 127.

Maadani, O., Shafiee, M., and Egorov, I., 2021: Climate Change Challenges for Flexible Pavement in Canada: An Overview. *Journal of Cold Regions Engineering*, 35(4), 03121002.

Megahed, Y., Shaker, A. and Yan, W.Y., 2021. Fusion of Airborne Lidar Point Clouds and Aerial Images for Heterogeneous Land-Use Urban Mapping. *Remote Sensing*, 13(4), p.814.

Megahed, Y., Yan, W.Y. and Shaker, A., 2021. Detection Of Urban Features by Multilevel Classification of Multispectral Airborne Lidar Data Fused with Very High-Resolution Images. *Journal of Applied Remote Sensing*, 15(4), pp.044521-044521.

Miller, J.S., Bellinger, W.Y., 2014: Distress Identification Manual for the Long-Term Pavement Performance Program (5th revised edition), Federal Highway Administration, Tech. Rep. FHWA-HRT-13-092, May 2014. [Online]. Available: https://www.fhwa.dot.gov/publications/research/infrastructure/pavements/ltpp/reports/03031/03031.pdf

Mohan, A. and Poobal, S., 2018. Crack Detection Using Image Processing: A Critical Review and Analysis. *Alexandria Engineering journal*, 57(2), pp. 787-798.

Munawar, H.S., Hammad, A.W., Haddad, A., Soares, C.A.P. and Waller, S.T., 2021. Image-Based Crack Detection Methods: A Review. *Infrastructures*, 6(8), p.115.

Nair, V., & Hinton, G. E. (2010). Rectified Linear Units Improve Restricted Boltzmann Machines. *In Proceedings of the 27th*

International Conference on machine learning (ICML-10) (pp. 807-814).

Nguyen, S.D., Tran, T.S., Tran, V.P., Lee, H.J., Piran, M.J., and Le, V.P., 2023: Deep Learning-Based Crack Detection: A Survey. *International Journal of Pavement Research and Technology*, 16(4), 943-967.

Office of the Auditor General of Ontario, 2016: Annual Report 2016, 3.10: Ministry of Transportation - Road Infrastructure Construction Contract Awarding and Oversight. [Online]. Available: https://www.auditor.on.ca/en/content/annualreports/arreports/en16/v1_310en16.pdf

Office of the Auditor General of Ontario, 2021: Inspection and Maintenance of the Province’s Bridges and Culverts. [Online]. Available: https://www.auditor.on.ca/en/content/annualreports/arreports/en21/ AR_InspectBridges_en21.pdf

Oliveira, H. and Correia, P.L., 2014, October. Crackit—An Image Processing Toolbox for Crack Detection and Characterization. *IEEE International Conference on Image Processing (ICIP)* (pp. 798-802).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. and Vanderplas, J., 2011. Scikit-Learn: Machine Learning in Python. *The Journal of Machine Learning Research*, 12, pp.2825-2830.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A., 2016: You only look once: Unified, real-time object detection in *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).

Ren, S., He, K., Girshick, R., and Sun, J., 2015: Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems,* 28.

Saberironaghi, A., and Ren, J., 2024: Depthcracknet: A Deep Learning Model for Automatic Pavement Crack Detection. *Journal of Imaging*, 10(5), 100.

Sabouri, M. and Sepidbar, A., 2023. SUT-Crack: A Comprehensive Dataset for Pavement Crack Detection Across All Methods. *Data in Brief*, 51, p.109642.

Shi, Y., Cui, L., Qi, Z., Meng, F. and Chen, Z., 2016. Automatic Road Crack Detection Using Random Structured Forests. *IEEE Transactions on Intelligent Transportation Systems*, 17(12), pp.3434-3445.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R., 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *The Journal of Machine Learning Research*, 15(1), pp.1929-1958.

Yan, W.Y., Shaker, A. and El-Ashmawy, N., 2015. Urban Land Cover Classification Using Airborne Lidar Data: A Review. *Remote Sensing of Environment*, 158, pp. 295-310.

Yang, B., Qin, Z., Zhou, Q., Li, H., Li, L., and Yang, X., 2020: Pavement damage behaviour of urban roads in seasonally frozen saline ground regions. *Cold Regions Science and Technology*, 174, 103035.

Yang, F., Zhang, L., Yu, S., Prokhorov, D., Mei, X., and Ling, H., 2019: Feature pyramid and hierarchical boosting network for pavement crack detection. *IEEE Transactions on Intelligent Transportation Systems*, 21(4), 1525-1535.

Zhang, A., Wang, K.C., Fei, Y., Liu, Y., Tao, S., Chen, C., Li, J.Q. and Li, B., 2018. Deep Learning–Based Fully Automated Pavement Crack Detection on 3D Asphalt Surfaces with An Improved Cracknet. *Journal of Computing in Civil Engineering*, 32(5), p.04018041.

Zhang, A., Wang, K.C., Li, B., Yang, E., Dai, X., Peng, Y., Fei, Y., Liu, Y., Li, J.Q. and Chen, C., 2017. Automated Pixel-Level Pavement Crack Detection On 3D Asphalt Surfaces Using a Deep-Learning Network. *Computer-Aided Civil and Infrastructure Engineering*, 32(10), pp.805-819.

Zhang, L., Yang, F., Zhang, Y. D., and Zhu, Y. J., 2016: Road Crack Detection Using a Deep Convolutional Neural Network. IEEE *International Conference on Image Processing (ICIP)* (pp. 3708-3712).

Zhang, X., Wang, H., Hsieh, Y.A., Yang, Z., Yezzi, A., and Tsai, Y.C., 2025: Deep Learning for Crack Detection: A Review of Learning Paradigms, Generalizability, and Datasets. *arXiv preprint arXiv*:2508.10256.

Zou, Q., Cao, Y., Li, Q., Mao, Q. and Wang, S., 2012. CrackTree: Automatic Crack Detection from Pavement Images. *Pattern Recognition Letters*, 33(3), pp.227-238.