CTIPS

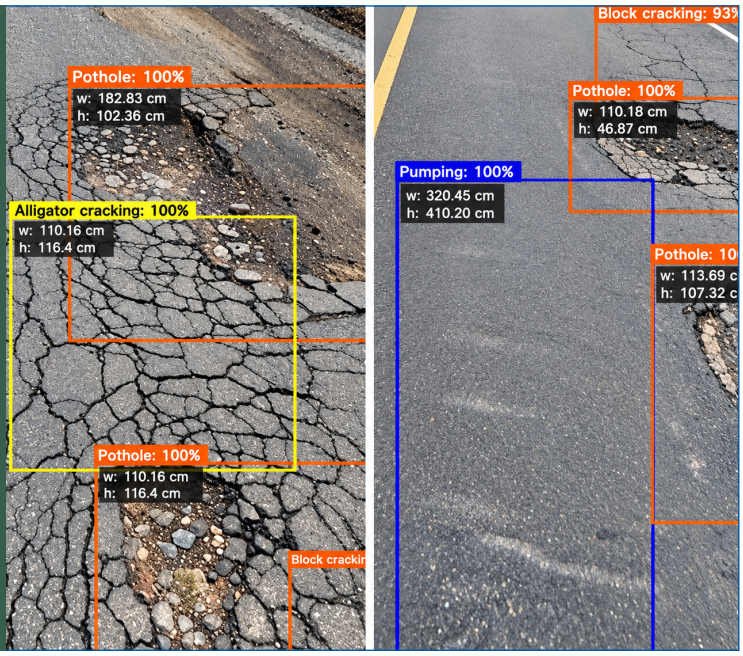
Center for Transformative Infrastructure Preservation and Sustainability

USDOT Region 8 University Transportation Center

## CTIPS-26-010

A. Aboah
D. Tolliver
B.A. Kyem
E. Denteh
J.K. Asamoah

# AN END-TO-END DEEP LEARNING SYSTEM FOR PAVEMENT DISTRESS DETECTION, SEVERITY ESTIMATION, AND CONDITION REPORTING




A University Transportation Center sponsored by the U.S. Department of Transportation serving the Center for Transformative Infrastructure Preservation and Sustainability members:

Colorado State University
Fort Lewis College
North Dakota State University
South Dakota State University
United Tribes Technical College
University of Colorado Denver
University of Denver
University of North Dakota
University of Utah
University of Wyoming
Utah State University

## Technical Report Documentation Page

| 1. Report No. | 2. Government Accession No. | 3. Recipient's Catalog No. |
|---|---|---|
| CTIPS-26-010 | | |

| 4. Title and Subtitle | 5. Report Date |
|---|---|
| An End-to-End Deep Learning System for Pavement Distress Detection, Severity Estimation, and Condition Reporting | October 2026 |
| | 6. Performing Organization Code: CTIPS |

| 7. Author(s) | 8. Performing Organization Report No. |
|---|---|
| • Armstrong Aboah https://orcid.org/0000-0002-1605-1545<br>• Denver Tolliver https://orcid.org/0000-0002-8522-9394<br>• Blessing Agyei Kyem https://orcid.org/0009-0006-6360-6386<br>• Eugene Denteh https://orcid.org/0009-0001-7597-6732<br>• Joshua Kofi Asamoah https://orcid.org/0009-0001-7597-6732 | CTIPS-26-010 |

| 9. Performing Organization Name and Address | 10. Work Unit No. |
|---|---|
| Center for Transformative Infrastructure Preservation and Sustainability<br>North Dakota State University<br>PO Box 6050<br>Fargo, ND 58108 | 11. Contract or Grant No.: 69A3552348308 |

| 12. Sponsoring Agency Name and Address | 13. Type of Report and Period Covered |
|---|---|
| Department of Transportation<br>Research and Innovative Technology Administration<br>400 7th Street, SW<br>Washington, DC 20590 | Final Report |
| | 14. Sponsoring Agency Code |

15. Supplementary Notes

Conducted in cooperation with the U.S. Department of Transportation, Federal Highway Administration.

16. Abstract

Regular assessment of pavement condition is critical for safe and functional roadways, yet traditional inspections are subjective, labor-intensive, and difficult to scale. Recent AI-based approaches have automated distress detection, but they produce only numerical outputs without the plain-language explanations needed to communicate findings to engineers, agencies, and the public. To address this gap, this project introduces PaveCap, a multimodal framework that assesses pavement condition through both numerical prediction and natural language description. The framework consists of two components: (1) a single-shot pavement condition index (PCI) estimation network that combines YOLOv8, the segment anything model (SAM), and a four-layer convolutional neural network to predict PCI values on a 0 to 100 scale; (2) a dense captioning network that uses a YOLOv8 backbone, a transformer architecture, and a convolutional feed-forward module to generate structured descriptions of pavement severity and distress conditions. To train and evaluate the framework, we developed a multimodal dataset of 7,158 pavement images with bounding box annotations, expert-authored textual descriptions, and PCI values aligned with ASTM D6433 guidelines. The PCI estimation network achieved a mean absolute error of 16.21, and the dense captioning network generated accurate descriptions with BLEU-1, GLEU, and METEOR scores of 0.3799, 0.3234, and 0.4253, outperforming five alternative backbones. Overall, PaveCap advances pavement condition assessment from numerical prediction to interpretable, human-centered AI, providing transportation agencies with a scalable tool that both quantifies pavement condition and explains it in natural language to support more transparent maintenance decisions.

| 17. Key Words | 18. Distribution Statement |
|---|---|
| Pavement Condition Index, Deep learning, Dense Captioning, Pavement Condition Assessment, | No restrictions. This document is available through the National Technical Information Service, Springfield, VA 22161. |

| 19. Security Classif. (of this report) | 20. Security Classif. (of this page) | 21. No. of Pages | 22. Price |
|---|---|---|---|
| Unclassified | Unclassified | 43 | n/a |

**Form DOT F 1700.7 (8-72)** **Reproduction of completed page authorized**




# An End-to-End Deep Learning System for Pavement Distress Detection, Severity Estimation, and Condition Reporting

Armstrong Aboah
Denver Tolliver
Blessing Agyei Kyem
Eugene Denteh
Joshua Kofi Asamoah

North Dakota State University
Fargo, ND 58105

October 2026

## About the Center for Transformative Infrastructure Preservation and Sustainability

The Center for Transformative Infrastructure Preservation and Sustainability (CTIPS) is the US DOT Region 8 University Transportation Center (UTC) located at North Dakota State University. CTIPS will focus on "Preserving the Existing Transportation System," along with consortia members Colorado State University, Fort Lewis College, South Dakota State University, United Tribes Technical College, University of Colorado Denver, University of Denver, University of North Dakota, University of Utah, University of Wyoming and Utah State University.


## Acknowledgements

The authors sincerely thank Upper Great Plains Transportation Institute (UGPTI) and The Center for Transformative Infrastructure Preservation and Sustainability (CTIPS) for the invaluable support and assistance.


## U.S. Department of Transportation (US DOT) Disclaimer

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated in the interest of information exchange. The report is funded, partially or entirely, by a grant from the U.S. Department of Transportation's University Transportation Centers Program. However, the U.S. Government assumes no liability for the contents or use thereof.


North Dakota State University does not discriminate in its programs and activities on the basis of age, color, gender expression/identity, genetic information, marital status, national origin, participation in lawful off-campus activity, physical or mental disability, pregnancy, public assistance status, race, religion, sex, sexual orientation, spousal relationship to current employee, or veteran status, as applicable. Direct inquiries to Vice Provost, Title IX/ADA Coordinator, Old Main 100, (701) 231-7708, ndsu.eoaa@ndsu.edu.

## ABSTRACT

Regular assessment of pavement condition is critical for safe and functional roadways, yet traditional inspections are subjective, labor-intensive, and difficult to scale. Recent AI-based approaches have automated distress detection, but they produce only numerical outputs without the plain-language explanations needed to communicate findings to engineers, agencies, and the public. To address this gap, this project introduces PaveCap, a multimodal framework that assesses pavement condition through both numerical prediction and natural language description. The framework consists of two components: (1) a single-shot pavement condition index (PCI) estimation network that combines YOLOv8, the segment anything model (SAM), and a four-layer convolutional neural network to predict PCI values on a 0 to 100 scale; (2) a dense captioning network that uses a YOLOv8 backbone, a transformer architecture, and a convolutional feed-forward module to generate structured descriptions of pavement severity and distress conditions. To train and evaluate the framework, we developed a multimodal dataset of 7,158 pavement images with bounding box annotations, expert-authored textual descriptions, and PCI values aligned with ASTM D6433 guidelines. The PCI estimation network achieved a mean absolute error of 16.21, and the dense captioning network generated accurate descriptions with BLEU-1, GLEU, and METEOR scores of 0.3799, 0.3234, and 0.4253, outperforming five alternative backbones. Overall, PaveCap advances pavement condition assessment from numerical prediction to interpretable, human-centered AI, providing transportation agencies with a scalable tool that both quantifies pavement condition and explains it in natural language to support more transparent maintenance decisions.

# CONTENTS


**1. INTRODUCTION AND BACKGROUND ..... 1**

1.1 Project Motivation ..... 1

1.2 Research, Objectives, and Tasks ..... 2

1.3 Report Overview ..... 3

**2. LITERATURE REVIEW ..... 4**

2.1 Deep Learning-Based PCI Estimation Methods ..... 4

2.2 Image Captioning ..... 5

**3. DATASET ..... 7**

3.1 Overview ..... 7

3.2 Bounding Box Annotations ..... 8

3.3 Textual Annotations (Captions) ..... 9

**4. METHODOLOGY ..... 10**

4.1 Problem Formulation and Overview ..... 10

4.2 PCI Estimation Network ..... 10

4.2.1 Accurate Object Detection Model (YOLOv8) ..... 11

4.2.2 Zero-Shot Object Segmentation Model ..... 12

4.2.3 Four-Layer Convolutional Neural Network ..... 12

4.3 Dense Captioning Network ..... 13

4.4 Evaluation Metrics ..... 15

4.4.1 PCI Estimation Network ..... 15

4.4.2 Dense Captioning Network ..... 16

4.5 Experimental Setup ..... 18

4.5.1 PCI Estimation Network ..... 18

4.5.2 Dense Captioning Network ..... 18

**5. RESULTS AND DISCUSSION ..... 20**

5.1 PCI Estimation Network ..... 20

5.1.1 Qualitative Results of the PCI Network ..... 20

5.1.2 Error Analysis of the Network ..... 22

5.2 Dense Captioning Network ..... 23

5.3 Ablation Studies ..... 26

5.3.1 PCI Estimation Network ..... 26

5.3.2 Dense Captioning Network ..... 27

**6. CONCLUSIONS AND RECOMMENDATIONS ..... 29**

**7. REFERENCES ..... 30**

## LIST OF FIGURES

Figure 1. Sample images from the dataset depicting various pavement conditions. ..... 8
Figure 2. Examples of pavement distress types with bounding box annotations. ..... 9
Figure 3. Example of a pavement image and its corresponding caption ..... 9
Figure 4. Overall two-stage framework for our dense pavement captioning network ..... 11
Figure 5. Exact PCI predictions for samples of images. ..... 20
Figure 6. Instances of slightly off PCI predictions. ..... 21
Figure 7. Instances of poor performance in PCI estimation. ..... 22
Figure 8. Error distribution in PCI estimation network. ..... 23
Figure 9. Frequency distribution and heatmap of prediction errors ..... 23
Figure 10. Examples of predicted accurate captions for sample pavement images. ..... 24
Figure 11. Instances of the dense captioning network's shortcomings. ..... 25
Figure 12. Model correcting annotation error. ..... 25
Figure 13. Overall framework for the dense captioning network (inference pipeline) ..... 26

## LIST OF TABLES

Table 1. Classification of PCI ranges ..... 7
Table 2. Hyperparameters and training settings for the PCI estimation network ..... 18
Table 3. Hyperparameters and training settings for the dense captioning network ..... 19
Table 4. Mean and standard deviation of evaluation metrics for dense captioning ..... 26
Table 5. Ablation study results for the PCI estimation network ..... 27
Table 6. Performance metrics for different backbones used to train the dense captioning network ..... 28

## LIST OF KEY TERMS

| | |
|---|---|
| AI | Artificial Intelligence |
| ASTM | American Society for Testing and Materials |
| BCE | Binary Cross-Entropy |
| BLEU | Bilingual Evaluation Understudy |
| BP | Brevity Penalty |
| CNN | Convolutional Neural Network |
| DSA | Doubly Stochastic Attention |
| FPS | Frames Per Second |
| GLEU | Google-BLEU |
| IoU | Intersection over Union |
| MAE | Mean Absolute Error |
| METEOR | Metric for Evaluation of Translation with Explicit ORdering |
| MSE | Mean Squared Error |
| NLP | Natural Language Processing |
| PCI | Pavement Condition Index |
| ReLU | Rectified Linear Unit |
| RT-DETR | Real-Time Detection Transformer |
| SAM | Segment Anything Model |
| YOLO | You Only Look Once |

## EXECUTIVE SUMMARY

Pavement condition assessment is essential for maintaining safe and functional roadways, yet traditional manual inspections are subjective, labor-intensive, and difficult to scale. Recent AI-based approaches have automated distress detection but treat the task as a unimodal problem, producing only numerical outputs without the human-readable explanations needed for transparent decision-making among engineers, policymakers, and the public.

To address this gap, this project introduces PaveCap, a multimodal framework that reformulates pavement condition assessment as a joint vision–language task. PaveCap integrates two complementary networks. The first is a single-shot pavement condition index (PCI) estimation network that combines YOLOv8 for distress detection, the segment anything model (SAM) for zero-shot segmentation, and a four-layer convolutional neural network that fuses the image with segmentation masks to regress the PCI on a 0–100 scale. The second is a dense captioning network that uses a frozen YOLOv8 backbone, a transformer encoder–decoder with a convolutional feed-forward module, and a language decoder to generate structured descriptions of pavement severity, present distresses, absent distresses, and PCI value. To support this framework, we developed a multimodal dataset of 7,158 top-down pavement images collected from Jefferson City, MO; Peoria, IL; and Washington, IL, extending the original PCI labels with bounding box annotations across six distress types and expert-authored textual descriptions aligned with ASTM D6433 guidelines. The dataset and framework are made publicly available.

PaveCap delivers strong performance across both tasks. The PCI estimation network achieves a validation mean absolute error of 16.21, with ablation results confirming that SAM-generated masks improve accuracy by providing complementary spatial cues. The dense captioning network achieves BLEU-1, GLEU, and METEOR scores of 0.3799, 0.3234, and 0.4253, respectively, outperforming five alternative backbones (YOLOv9, YOLOv10, YOLOv11, YOLOv12, and RT-DETR). Qualitatively, the model generates accurate captions and, in some cases, corrects annotation errors in the ground truth. Overall, PaveCap advances pavement assessment from numerical prediction toward interpretable, human-centered AI, offering transportation agencies a scalable tool that both quantifies pavement condition and explains it in natural language, thereby supporting more transparent maintenance decisions and more efficient use of infrastructure resources. The underlying pavement image and PCI dataset is publicly available at https://github.com/UM-Titan/DSPS24.

# 1. INTRODUCTION AND BACKGROUND

## 1.1 Project Motivation

Pavement condition assessment is crucial for evaluating the service conditions of road infrastructure. This process facilitates the early detection of structural issues, playing a pivotal role in maintaining serviceability and ensuring pavement longevity. Traditionally, experienced engineers conduct visual inspections to assess pavement conditions, focusing primarily on pavement distresses found on the pavement's surface. While this approach provides a detailed assessment of the pavement condition, this method suffers from inefficiency and subjectivity due to the variability in inspectors' experience, highlighting the need for an automated and consistent assessment process.

In recent years, the advent of digital image-based pavement assessment has accelerated the automated assessment process, thereby mitigating subjectivity in analytical processes traditionally performed by engineers. Early research in this area employed conventional image processing techniques, such as threshold segmentation, edge detection, and morphological operations, to identify pavement distress patterns (Hoang and Nguyen 2019). Despite their foundational significance, these methods faced limitations in addressing the complexity and variability of real-world pavement conditions, necessitating more advanced approaches (Chavan et al. 2022). More recently, the emergence of machine learning brought significant improvements, mainly through feature engineering and the use of support vector machines for distress classification (Hoang and Nguyen 2019). Despite these advancements, a significant paradigm shift occurred with the introduction of deep learning, which greatly enhanced the performance and robustness of computer vision algorithms. Today, deep learning-based object classification algorithms are extensively used for categorizing pavement distress, while object detection and segmentation algorithms accurately identify damaged areas (Majidifard et al. 2020a; Qureshi et al. 2022).

Despite the numerous studies demonstrating the efficacy of deep learning models in identifying pavement distress, a significant challenge persists—pavement condition assessment remains a multimodal problem, but it is often oversimplified as a unimodal problem. That is, current approaches predominantly focus on numerical results derived from image classification, detection, or segmentation, which fail to fully capture the nuanced understanding necessary for effective communication and decision-making (Karballaeezadeh et al. 2020). A highly abstract natural language description can more comprehensively summarize and convey the pavement assessment process and results to human stakeholders (Han et al. 2023). This gap in current research underscores the urgent need for innovative methodologies that incorporate natural language as a modality in assessing pavement conditions. Therefore, this study seeks to pioneer the use of vision-language models for dense pavement captioning, transforming the way pavement conditions are assessed and reported.

The emergence of dense captioning models (DCMs) has enabled joint image–text embeddings, allowing textual descriptions, a natural way for humans to understand scenes, to be incorporated into the analysis. DCMs have demonstrated significant success in other domains by providing accurate assessments and generating human-understandable explanations for their predictions.

This study seeks to explore the application of DCMs in pavement condition assessment, aiming to develop a model that can process pavement images and produce a natural language description of the visual representation. Our proposed model would also provide justification for captioning through bounding box grounding and ultimately offer a pavement condition index (PCI) value for the given image.

## 1.2 Research, Objectives, and Tasks

To address the limitations of existing unimodal pavement assessment approaches, this project aims to develop a multimodal framework that jointly predicts PCI values and generates natural language descriptions of pavement condition. Specifically, we seek to construct a richly annotated multimodal dataset, design a single-shot PCI estimation architecture, develop a dense captioning network for interpretable outputs, and validate the framework through rigorous evaluation and ablation studies. By advancing both the technical and communicative capabilities of automated pavement assessment, we aim to support more transparent maintenance decisions and more efficient use of infrastructure resources. The following research tasks are proposed to achieve these objectives:

**Task 1: Multimodal Dataset Development**

Extend an existing pavement image dataset of 7,158 top-down images collected from three U.S. cities (Jefferson City, MO; Peoria, IL; and Washington, IL) by manually adding bounding box annotations across six distress types (longitudinal, transverse, block, diagonal, and alligator cracking, as well as potholes and patching) and expert-authored textual descriptions aligned with ASTM D6433 guidelines.

**Task 2: Development of a Single-Shot PCI Estimation Network**

Design and implement a PCI estimation architecture that combines You Only Look Once version 8 (YOLOv8) for distress detection, the segment anything model (SAM) for zero-shot segmentation, and a four-layer convolutional neural network that fuses the raw image with SAM-generated binary masks to predict PCI values on a 0 to 100 scale.

**Task 3: Development of a Dense Captioning Network**

Design and implement a dense captioning network that uses a frozen YOLOv8 backbone, a transformer encoder–decoder enhanced with a convolutional feed-forward module, and a language decoder to generate structured natural language descriptions of pavement severity, present distresses, and absent distresses.

**Task 4: Model Validation and Ablation Studies**

Evaluate both networks using established metrics, including mean absolute error (MAE) for PCI estimation and BLEU, GLEU, and METEOR scores for dense captioning. Conduct ablation studies to quantify the contribution of the segmentation component to PCI estimation and to compare alternative backbone architectures (YOLOv9, YOLOv10, YOLOv11, YOLOv12, and RT-DETR) for the captioning network.

**Task 5: Framework Integration and Qualitative Analysis**

Integrate both networks into a unified inference pipeline that processes a pavement image and produces a combined output containing the predicted PCI value and a detailed natural language description. Conduct qualitative analysis and error diagnostics to identify failure modes and opportunities for future refinement.

## 1.3 Report Overview

This report is organized to provide a clear understanding of the background, methodology, and outcomes. The remainder of the report is structured as follows. Section 2 reviews related work on deep learning-based PCI estimation and image captioning. Section 3 describes the dataset and annotation procedures. Section 4 presents the proposed methodology, network architectures, evaluation metrics, and experimental setup. Section 5 discusses the results, including qualitative evaluations, error analysis, and ablation studies. Section 6 summarizes the conclusions and recommendations for future research.

# 2. LITERATURE REVIEW

## 2.1 Deep Learning-Based PCI Estimation Methods

The PCI is a critical metric for assessing road surface quality, essential for effective infrastructure management (Karim et al. 2016; Zafar et al. 2019; Danyo et al. 2025). Traditional methods of PCI estimation, which rely on manual inspections, are often criticized for their subjectivity and inefficiency. To address these limitations, researchers have increasingly turned to deep learning techniques to automate PCI assessments and improve accuracy.

Early advancements in this field were made by Majidifard et al. (2020a), who proposed a hybrid model integrating YOLO for object detection with U-Net for image segmentation to classify and quantify pavement distresses. This approach provided a comprehensive assessment of pavement conditions but required large annotated datasets and significant computational power (Agyei Kyem et al. 2026). These requirements, coupled with the challenges of ensuring high image quality, highlighted the need for further refinement in deep learning models for PCI estimation.

Sirhan et al. (2022) trained a deep neural network on 671,060 pavement distress observations to predict PCI values, achieving a root mean squared error (MSE) of 1.876 and significantly outperforming both linear and nonlinear regression benchmarks. They also identified alligator cracking, swelling, rutting, and potholes as the most influential distress features. Following this, Han et al. (2023) utilized aerial imagery combined with convolutional neural networks (CNNs) to evaluate pavement conditions. Their proposed model (PCIer) demonstrated high accuracy in categorizing pavement conditions across large areas. However, this approach faced challenges related to the need for specialized aerial equipment and the potential degradation of image quality due to environmental factors. Additionally, capturing fine-grained details essential for accurate PCI estimation proved difficult with aerial imagery.

In another significant development, Owor et al. (2023) developed a multitask learning framework that combines detection, segmentation, and PCI estimation into a single model. This unified model enhanced the efficiency of PCI assessments while maintaining accuracy across tasks. However, the increased computational complexity and the potential for performance tradeoffs between tasks posed significant challenges, limiting the framework's scalability.

Building on these efforts, Ibragimov et al. (2024) introduced an automated system that uses deep learning for crack detection, followed by image processing algorithms for crack width estimation. While this method significantly improved the speed and accuracy of PCI assessments, it was heavily dependent on high-quality images and required substantial computational resources, limiting its applicability in varied environments.

Despite these advancements, current deep learning approaches for PCI estimation face several limitations, including dependency on high-quality data, substantial computational demands, and challenges in generalizing across diverse pavement conditions. These challenges underscore the necessity for developing a more streamlined and efficient PCI estimation framework that can operate effectively with diverse input data and reduced computational requirements. Such a

framework would advance the field by addressing the current limitations and improving the scalability and practicality of deep learning-based PCI assessment tools.

## 2.2 Image Captioning

Image captioning is a process that generates a textual description for an image using natural language processing and computer vision techniques to automatically create descriptions of image content in natural language (Karthik et al. 2024). Conventionally, image captioning relied on models such as CNN-LSTM, which had limitations in capturing complex relationships between visual elements and textual descriptions (Kushal et al. 2024). However, image captioning has not yet been applied in the pavement engineering domain. Biradar et al. (2023) used a CNN-LSTM model for automatic image captioning, aiming to better capture the link between images and text. However, their model struggled with identifying key objects and their relationships in images and generating grammatically and logically sound captions. Gowda et al. (2024) proposed "captionify," a CNN-LSTM model that outperformed existing approaches in generating accurate captions by combining CNNs' visual feature extraction with long short-term memory's (LSTM's) textual generation, enhancing accuracy and context relevance in image captioning. Lala and Kumar (2022) developed an image captioning model using a modified Xception-based CNN and LSTM. While their model showed improved performance on datasets such as Flickr8k, it struggled with understanding the semantics of complex scenes, potentially impacting caption quality for images. To address the aforementioned limitations, Yan et al. (2022) proposed a task-adaptive attention module to improve image captioning by balancing visual and non-visual word generation. While showing promise, the module's complexity and potential for further optimization required further investigation. Despite these improvements, attention-based LSTMs still struggled with the inherent sequential processing constraints of recurrent neural networks (RNNs), which limited their ability to parallelize computations and handle long-range dependencies efficiently (Mishra et al. 2021).

The introduction of transformer architectures marked a significant leap forward in image captioning. Transformers, which rely on self-attention mechanisms, can process sequences in parallel, thereby overcoming the sequential bottlenecks of RNNs. This architecture allows for better handling of long-range dependencies and more efficient computation (Kandala et al. 2022; Wang et al. 2020). Yu et al. (2020) proposed a multimodal transformer (MT) model for image captioning, utilizing multiview visual features and a unified attention block to capture relationships within and between image and text data. Despite outperforming previous methods on the MSCOCO dataset, the model's complexity and computational cost remain limitations. Addressing the limitations of the MT model, Shao et al. (2022) proposed a transformer-based dense image captioning architecture (TDC) that prioritizes more informative regions using a region-object correlation score unit (ROCSU). This method overcomes the vulnerability of LSTM in handling long sequences and the equal importance given to all regions in previous models. Despite its improvements, the TDC model still struggled with the computational overhead associated with the ROCSU. These studies suggest that transformer architectures significantly improve image captioning performance by capturing complex interactions and prioritizing informative regions, although they often face challenges related to computational complexity (Kyem et al. 2025).

In addition to the transformer architecture, convolutional feed-forward networks have also been integrated into image captioning models to enhance performance. These networks help refine the visual features extracted by the CNN, ensuring that the most critical information is retained and nonessential details are filtered out (Liu et al. 2021). This combination of convolutional and transformer-based approaches has been shown to significantly outperform traditional methods (Wang et al. 2020).

# 3. DATASET

## 3.1 Overview

The dataset used for this research consists of top-down views of pavement image data and only its associated PCI (Gyamfi et al. 2024). The images were acquired under varying conditions from different sources in three cities: Jefferson City, MO; Peoria, IL; and Washington, IL. Figure 1 shows samples of these pavement images with their corresponding PCI displayed on the top. These images encompassed a diverse range of pavement conditions, including various types of distresses such as cracks, patching, and potholes. All PCI ratings were assigned by trained pavement engineering experts using standardized ASTM guidelines (ASTM 2024) to ensure the reliability and consistency of the ground truth labels. ASTM D6433 defines PCI at the pavement sample-unit level. In this study, each top-down image was treated as a localized pavement observation, and its label represents an image-level PCI estimate derived from the visible distress type, severity, and extent using ASTM-based principles. The PCI values associated with each image ranged from 0 to 100, where 0 represents the worst pavement condition and 100 represents the optimal condition. The original dataset contained only pavement images and their associated PCI values without any additional labels or annotations. For the purposes of this study, we extended the dataset by manually adding bounding box annotations and textual descriptions of the pavement condition to support object detection and caption generation tasks. Table 1 provides the ASTM-proposed standards for classifying PCI based on the distress type, severity, and extent. The following subsections provide details about these additional data annotation processes. Overall, the dataset consists of 7,158 images, which is split into training and validation using a split ratio of 80:20.

**Table 1.** Classification of PCI ranges

| **PCI Range** | **Class** |
|---|---|
| 0–11 | Failed |
| 11–25 | Serious |
| 26–40 | Very Poor |
| 41–55 | Poor |
| 56–70 | Fair |
| 71–85 | Satisfactory |
| 86–100 | Good |

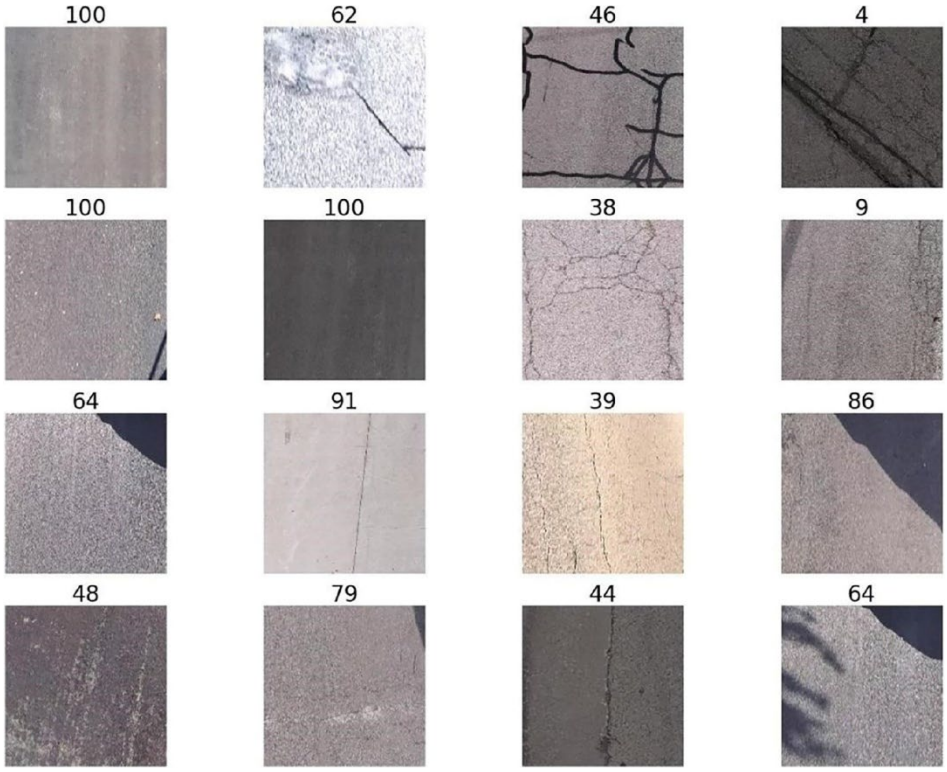



**Figure 1.** Sample images from the dataset depicting various pavement conditions.

## 3.2 Bounding Box Annotations

To enhance the dataset for our multimodal approach, we conducted additional annotations on the pavement images using the Computer Vision Annotation Tool version 1.1.0. This process involved creating bounding boxes around specific distress types. Seven primary distress types were considered for annotation: alligator cracking, longitudinal cracking, transverse cracking, potholes, patching, block cracking, and diagonal cracking.

To ensure annotation quality and consistency, we developed a set of guidelines for the annotators, including minimum size thresholds for each distress type, rules for handling overlapping distresses, and criteria for distinguishing between similar distress types. Figure 2 shows samples of pavement images with various types of cracks and patching.

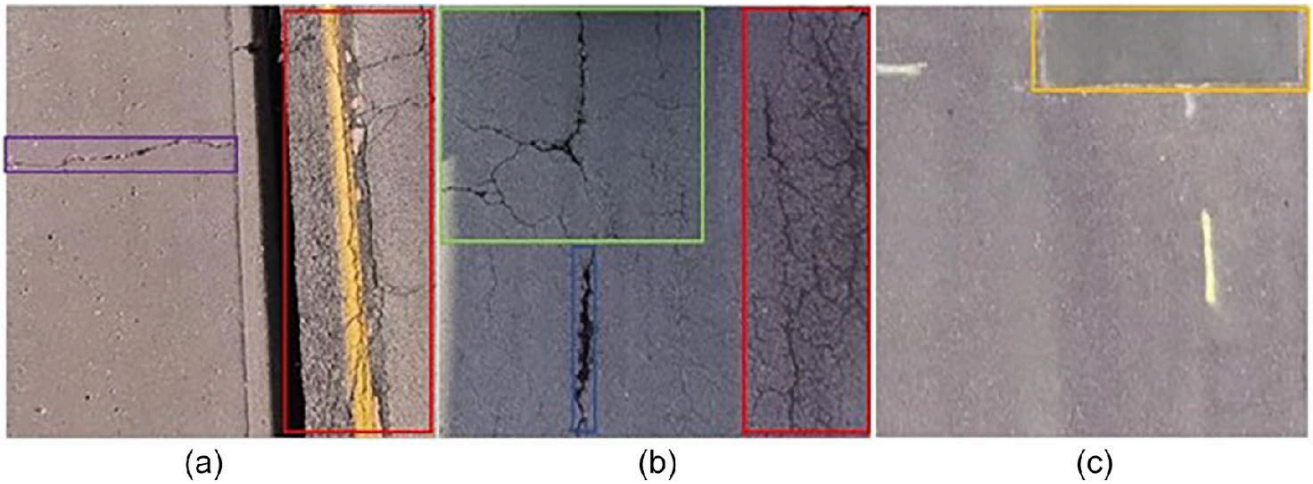



**Figure 2.** Examples of pavement distress types with bounding box annotations.

## 3.3 Textual Annotations (Captions)

To complement the bounding box annotations and PCI values, we developed a comprehensive textual annotation scheme for each pavement image. Figure 3 provides details about the textual annotation of a pavement image. This scheme was designed to provide a detailed and structured account of the pavement condition, significantly enhancing the dataset's richness and facilitating the training of our multimodal model. The textual annotations were meticulously crafted by pavement engineering experts to ensure accuracy and consistency across the dataset, following a standardized format comprising four key components. The first component, pavement severity, offers an overall assessment of the pavement's condition, categorized as high, medium, or low severity based on the extent and nature of observed distresses. This is followed by a detailed list of present distresses, which include specific types such as alligator cracking, longitudinal cracking, transverse cracking, potholes, and patching, along with their respective severity levels. To provide a comprehensive assessment, the third component explicitly mentions distress types not observed in the image, offering crucial negative information that aids in distinguishing between present and absent features. The final component incorporates the numerical value, providing a quantitative measure of the overall pavement condition.

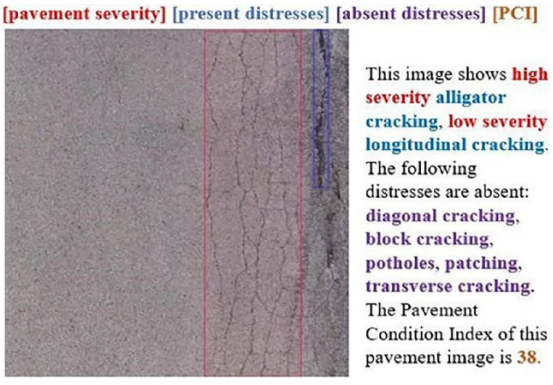



**Figure 3**. Example of a pavement image and its corresponding caption.

# 4. METHODOLOGY

## 4.1 Problem Formulation and Overview

Machine learning and deep learning approaches for pavement condition assessment have predominantly been treated as a unimodal problem (Tong et al. 2020; Zakeri et al. 2017; Majidifard et al. 2020b; Du et al. 2021; Nie and Wang 2018; Osman et al. 2021; Kyem et al. 2024; Yi et al. 2023; Owor et al. 2023; Mandal et al. 2018; Stricker et al. 2021; Han et al. 2022; Agyei Kyem et al. 2025) instead of a multimodal problem, relying heavily on only image-based analysis for distress detection, segmentation, and classification. In this study, we formulate the problem as a multi-modal task.

Let $I \in R^{H\times W\times C}$ represent a pavement image $H$ height, $W$ width, and $C$ channels. The goal is to estimate the PCI value from 0 to 100 and generate a dense caption that accurately describes the pavement condition. Our framework consists of two main components: a PCI prediction network and a dense captioning network. The PCI prediction network $f_{\text{PCI}}: R^{H\times W\times C} \rightarrow [0,100]$ is a neural network model with parameters $\theta$ designed to predict the PCI value from the input image $I$. The model $f_{PCI}$ can be expressed as $\hat{y} = f_{\text{PCI}}(I;\theta)$, where $\hat{y} \in [0,100]$ is the predicted PCI value. The dense captioning network $f_{\text{CAP}}: R^{H\times W\times C} \rightarrow \mathcal{T}$ is a model with parameters $\emptyset$ that outputs a set of textual descriptions $\mathcal{T} = \{t_1, t_2, \dots, t_n\}$ where $t_i$ represents individual textual descriptions. The model $f_{CAP}$ can be expressed as $\mathcal{T} = f_{\text{CAP}}(I;\theta)$. To generate the overall dense caption, the PCI prediction $\hat{y}$ and the textual descriptions $\mathcal{T}$ are combined. Let $f_{\text{final}}: [0,100] \times \mathcal{T} \rightarrow \mathcal{H}$ be a function with parameters $\psi$ that combines the PCI value and the set of textual descriptions to produce the overall dense caption $\mathcal{H}$. The overall dense caption $\mathcal{H} = f_{\text{final}}(\hat{y}, \mathcal{T}; \psi)$.

Our overall framework involves a series of steps. First, the pavement image is passed through a PCI estimation network to predict the PCI. Next, the same pavement image is passed through a dense captioning network to generate a dense caption. The results from both networks are combined to generate the final caption. Figure 4 illustrates the overall framework for our dense pavement captioning network. The next section goes into detail about the two main components of the framework.

## 4.2 PCI Estimation Network

We introduce a framework for PCI estimation, characterized by three core components: (1) a state-of-the-art object detection model; (2) a zero-shot object segmentation model; and (3) a four-layer convolutional neural network. The framework takes in pavement images processed through the YOLOv8 object detection model, which yields a set of bounding boxes and their associated classes. These bounding boxes, along with the input images, are subsequently passed to a zero-shot object segmentation model (SAM), which outputs binary masks using the bounding boxes as prompts. The resulting binary mask is then concatenated with the input image and fed into the four-layer convolutional neural network, ultimately producing the final PCI value. The framework of the PCI estimation network is illustrated in Figure 4. The three core components have subsequently been expanded.

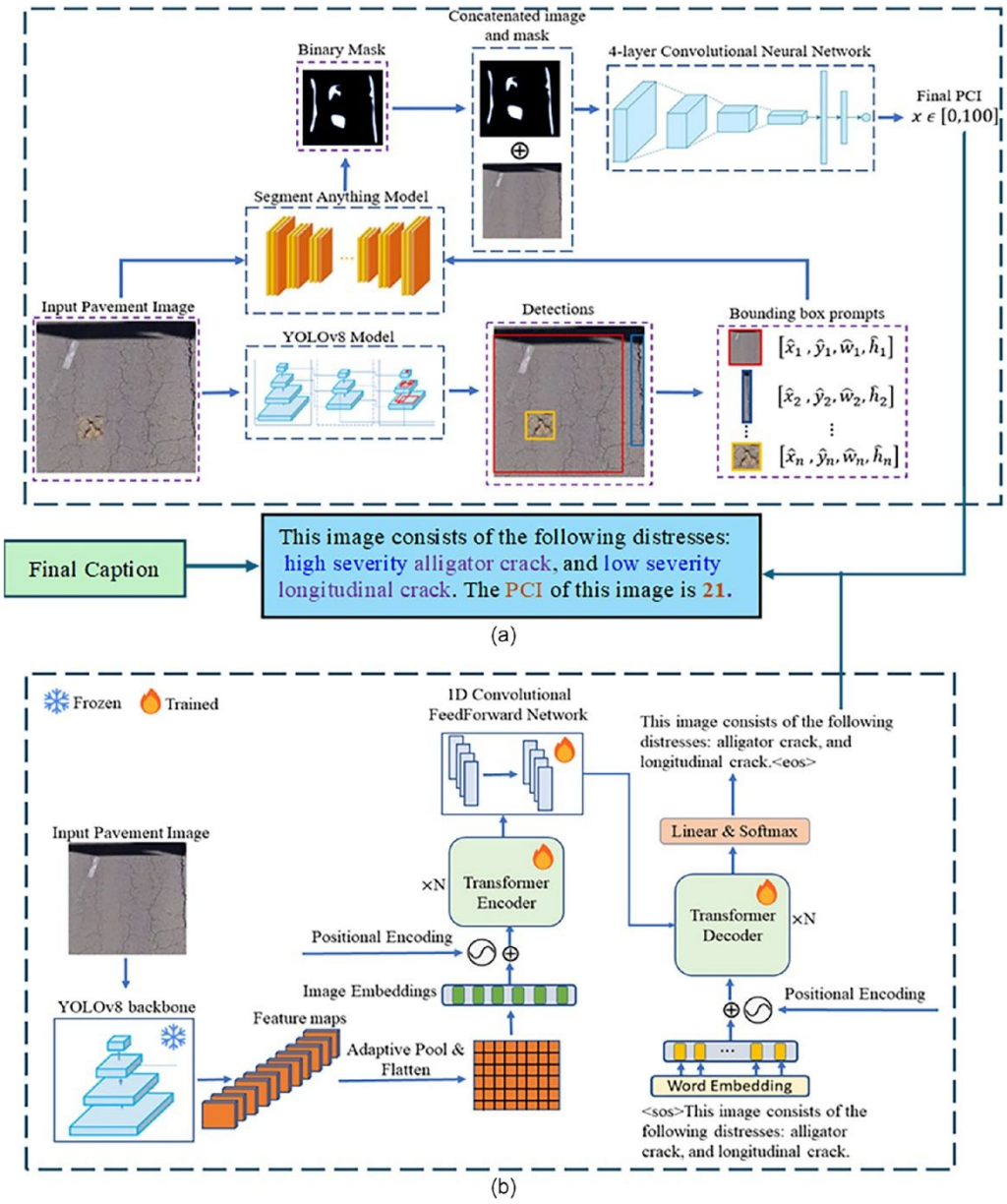



**Figure 4.** Overall two-stage framework for our dense pavement captioning network.

## Accurate Object Detection Model (YOLOv8)

In the initial phase of our framework, we utilize the medium size model of You Only Look Once version 8 (YOLOv8) object detection model to identify and classify objects within pavement images. YOLOv8 is known for its efficiency and accuracy, performing object detection in a single forward pass. The model operates on an input image I of dimensions H × W × C, where $H$ is the height, $W$ is the width, and $C$ is the number of channels. The YOLOv8 model processes the image through a series of convolutional layers to extract hierarchical features, resulting in feature maps of varying spatial resolutions. The detection head of YOLOv8 predicts bounding boxes and class probabilities for each object detected in the image. The bounding box for object $i$ is predicted as:

$$\widehat{B_i} = \left(\widehat{x_i}, \widehat{y_i}, \widehat{w_i}, \widehat{h_i}\right) \quad (1)$$

where $\widehat{x_i}$ and $\widehat{y_i}$ = center coordinates; and $\widehat{w_i}$ and $\widehat{h_i}$ = width and height of the bounding box, respectively. The class probability for object is computed as:

$$\hat{P}(\, c_i \mid I\,) = \frac{\exp(C_i \cdot f(I))}{\sum_j \exp(C_i \cdot f(I))} \quad (2)$$

Where $C_i$ = weight vector for class $i$; and $f(I)$ = feature map extracted from the image. The overall loss function of YOLOv8 combines localization loss, confidence loss, and classification loss:

$$\mathcal{L}_{\text{YOLO}} = \lambda_{\text{coord}} \sum_{i=1}^{N} L_{\text{loc}}(\widehat{B_i}, B_i) + \lambda_{\text{conf}} \sum_{i=1}^{N} L_{\text{conf}}(\widehat{P_{\text{conf},i}}, P_{\text{conf},i}) + \lambda_{\text{cls}} \sum_{i=1}^{N} L_{\text{cls}}\left(\hat{P}(\, c_i|I\,), P(\, c_i|I\,)\right) \quad (3)$$

where N = number of detected objects.

## Zero-Shot Object Segmentation Model

Following object detection, the framework advances to the segmentation stage, employing SAM ViT-B to generate precise binary masks for the detected objects. This SAM model leverages a zero-shot learning approach, meaning it can generate accurate segmentations without prior task-specific training. Given an input image $I$ of dimensions $H \times W \times C$ and bounding boxes $\widehat{B_i}$ produced by YOLOv8, SAM generates binary masks $\widehat{M_i}$ for each detected object. The SAM model utilizes the bounding boxes as prompts to refine and produce binary segmentation masks:

$$\widehat{M_i} = \text{SAM}\left(I, \widehat{B_i}\right) \quad (4)$$

where $\widehat{M_i}$ is the binary mask for object i generated by SAM. The segmentation process involves feature extraction from the input image, integration of the bounding box prompts, and mask generation. The SAM model's loss function combines dice loss and binary cross-entropy loss:

$$\mathcal{L}_{SAM} = \lambda_{dice}\mathcal{L}_{dice}\left(\widehat{M_i}, M_i\right) + \lambda_{bce}\mathcal{L}_{bce}\left(\widehat{M_i}, M_i\right) \quad (5)$$

where $M_i$ = ground truth mask for object $i$

## Four-Layer Convolutional Neural Network

The final stage of the framework involves a four-layer CNN designed to estimate the PCI value from the processed image data. The input to this network is a concatenation of the original pavement image and the binary masks generated by SAM, forming an input of dimensions $H \times W \times (C + 1)$. By concatenating the original pavement image with the binary masks, we create a composite input that retains both the raw visual details and the precise segmentation information. This concatenated input $I_{concat} = \left[I; \widehat{M_i}\right]$ is fed into the CNN, which consists of four convolutional layers. Each layer extracts and processes features, ultimately producing the final PCI value. The operation of each convolutional layer can be described as

$$O_i = \sigma(W_i * I_i + b_i)$$
(6)

where $O_i$ is the output feature map of the i-th convolutional layer, $W_i$ represents the convolutional kernel weights, $b_i$ represents the bias term, $I_i$ is the input feature map to the ith convolutional layer, σ denotes the activation function typically ReLU, and * denotes the convolution operation. The final convolutional layer outputs a feature map $O_4$ that is flattened and passed through a fully connected layer to map the extracted features to the PCI value:

$$PCI = f_{CNN}(O_4) \tag{7}$$

where $f_{CNN}$ is the function representing the fully connected layer. The loss function for the CNN is the MSE between the predicted PCI and the ground-truth PCI.

$$\mathcal{L}_{PCI} = \frac{1}{N}\sum_{i=1}^{N}\left(PCI_i - \widehat{PCI_i}\right)^2 \tag{8}$$

The final PCI value is added to the output of the textual descriptions generated by the dense captioning network. The next section talks about the components of the dense captioning network.

## 4.3 Dense Captioning Network

The study designed a dense captioning framework, composed of three main components: the YOLOv8 backbone network, a transformer encoder–decoder architecture, and a convolutional feed-forward module. The process begins with comprehensive feature extraction through the YOLOv8 backbone network, trained on a large dataset of pavement images, resulting in detailed feature maps. These feature maps are then standardized via an adaptive pooling layer and reshaped into a one-dimensional feature vector. This vector is fed into an enhanced transformer encoder–decoder architecture, where the encoder is augmented with a convolutional feed-forward layer consisting of two convolutional layers with a ReLU activation function, allowing for the extraction of intricate features. The encoded representations are concatenated with ground truth captions and processed through the transformer's decoder blocks. The final output layer generates a probability distribution over the vocabulary for each token position, conditioned on the previously generated tokens. The framework of the dense captioning network is illustrated in Figure 4.

Our dense captioning framework involves utilizing the YOLOv8 backbone network for comprehensive feature extraction. YOLOv8, trained on a substantial dataset of pavement images, is adept at capturing essential visual details. Importantly, the backbone of YOLOv8 is frozen during training to retain its learned representations. The process starts with an input image I of dimensions $H \times W \times C$, where H is the height, W is the width, and C represents the color channels. The YOLOv8 model processes this image to generate feature maps:

$$F = YOLOv8_{backbone}(I) \tag{9}$$

These feature maps F, which encapsulate detailed visual information, are then standardized to a fixed size using an adaptive pooling layer. To ensure uniformity in subsequent processing steps,

the feature maps F are passed through an adaptive pooling layer. This layer resizes the feature maps to a predetermined size $H_p \times W_p$, making them suitable for further operations. The adaptive pooling operation is defined as:

$$F_{pooled}(i,j) = \max_{m,n} F\left(\left\lfloor \frac{m \cdot H}{H_p} \right\rfloor, \left\lfloor \frac{n \cdot W}{W_p} \right\rfloor\right) \tag{10}$$

where $(i,j)$ are the corresponding coordinates in the pooled feature map, and $(m,n)$ are adjusted according to the pooling window. Following the adaptive pooling layers, $F_{pooled}$ is flattened and reshaped into a one-dimensional feature vector. This transformation prepares the data for input into the transformer encoder-decoder architecture. The process can be represented as:

$$f_{vector}(k) = F_{pooled}\left(\left\lfloor \frac{k}{W_p \cdot C} \right\rfloor, \left\lfloor \frac{k \bmod (W_p \cdot C)}{C} \right\rfloor, k \bmod C\right) \tag{11}$$

where k ranges from 0 to $H_p \cdot W_p \cdot C - 1$.

$$E = \text{Encoder}(f_{\text{vector}}) \tag{12}$$

Within the transformer encoder, the self-attention mechanism allows the model to weigh the importance of different parts of the pavement input image sequence:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{13}$$

where Q represents the queries, K represents the keys, V represents the values, and $d_k$ is the dimensionality of the keys. These vectors are derived from the input embeddings and allow the model to learn relationships among different parts of the visual feature sequence.

$$E_{\text{conv}} = W_2 * \sigma(W_1 * E + b_1) + b_2 \tag{14}$$

where σ is the ReLU activation function, $W_1$ and $W_2$ are convolutional kernel weights, and $b_1$ and $b_2$ are bias terms. The encoded representations $E_{conv}$ are concatenated with the ground-truth caption sequences $C_{gt}$ and passed into the transformer decoder. The concatenation step ensures that the model has access to both visual and textual information.

$$D_{\text{input}} = \left[E_{\text{conv}}; C_{\text{gt}}\right] \tag{15}$$

The transformer decoder processes these inputs sequentially to generate dense captions:

$$D = \text{Decoder}\left(D_{\text{input}}\right) \tag{16}$$

The final stage involves generating a probability distribution over the vocabulary for each token position using a fully connected layer. This layer produces unnormalized logits, which are then normalized using the softmax function.

$$z_i = W_o \cdot D_i + b_o \tag{17}$$

where $W_o$ is the learned weight matrix of the final fully connected output layer, $D_i$ is the hidden state vector of the transformer decoder at time step i, and $b_o$ is the output bias term.

$$P(t_i \mid t_{1:i-1}) = \frac{\exp(z_i)}{\sum_{j=1}^{V} \exp(z_j)} \tag{18}$$

where $P(t_i \mid t_{1:i-1})$ is the probability of token $t_i$ given the sequence of previously generated tokens $t_{1:i-1}$. The training of the model incorporates a combination of cross-entropy loss and doubly stochastic attention regularization. The cross-entropy loss $L_{CE}$ is given by:

$$L_{CE} = -\sum_{t=1}^{T} \log\big(P(t_t \mid t_{1:t-1})\big) \tag{19}$$

where T is the length of the sequence. To encourage the model to distribute its attention across the entire image rather than focusing on specific parts, we add a doubly stochastic attention regularization term $L_{DSA}$.

$$L_{DSA} = \sum_{l=1}^{L} \frac{1}{L} \Big(\sum_{d=1}^{D} \sum_{i=1}^{P^2} (1 - \sum_{c=1}^{T} \alpha_{cidl})^2\Big) \tag{20}$$

where D is the number of attention heads, L is the number of layers, P is the size of the feature map, and $\alpha_{cidl}$ is the attention weight. The total loss then becomes:

$$L = -\sum_{t=1}^{T} \log\big(P(t_t \mid t_{1:t-1})\big) + \sum_{l=1}^{L} \frac{1}{L} \Big(\sum_{d=1}^{D} \sum_{i=1}^{P^2} (1 - \sum_{c=1}^{T} \alpha_{cidl})^2\Big) \tag{21}$$

## 4.4 Evaluation Metrics

### PCI Estimation Network

In the evaluation of the PCI estimation network, we employed two primary metrics: MSE and MAE. These metrics were chosen for their effectiveness in quantifying the accuracy of continuous predictions, particularly in the context of regression tasks like PCI estimation.

**Mean Squared Error**

MSE is a widely used metric that measures the average of the squares of the errors, i.e., the average squared difference between the estimated values and the actual value.

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (\widehat{y_i} - y_i)^2 \tag{22}$$

where n is the number of data points, $\widehat{y_i}$ is the predicted PCI value for the i-th data point, and $y_i$ is the actual PCI value for the i-th data point.

**Mean Absolute Error**

MAE measures the average of the absolute differences between the predicted values and the actual values. The MAE is computed as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\widehat{y_i} - y_i| \tag{23}$$

where n is the number of data points, $\widehat{y_i}$ is the predicted PCI value for the i-th data point, and $y_i$ is the actual PCI value for the $i - th$ data point.

## Dense Captioning Network

For evaluating the dense captioning network, we utilized three well-established metrics: bilingual evaluation understudy (BLEU), the metric for evaluation of translation with explicit ordering (METEOR), and general language understudy (GLEU). These metrics are widely recognized for their ability to measure the quality of generated text against reference texts, making them particularly suitable for assessing the performance of captioning models.

**BLEU**

BLEU is a precision-based metric that evaluates the extent to which n-grams in the generated text match the n-grams in the reference text. This metric is particularly effective for machine translation and text generation tasks. The calculation of the BLEU score involves several steps. First, the n-gram precision is computed as follows:

$$P_n = \frac{\sum_{count}\min\left(count_{clip}(n\text{-}gram), count_{ref}(n\text{-}gram)\right)}{\sum_{count} count(n\text{-}gram)} \tag{24}$$

where $P_n$ is the precision for n-grams, $count_{clip}$ is the clipped count of matching n-grams, and $count_{ref}$ is the count of n-grams in the reference text. The brevity penalty (BP) is then calculated to account for the length of the generated text relative to the reference text.

$$BP = \begin{cases} 1, & if\ c > r \\ e^{\left(1-\frac{r}{c}\right)}, & if\ c \leq r \end{cases} \tag{25}$$

where c is the length of the candidate text, and r is the length of the reference text. The final BLEU score is computed as:

$$BLEU = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log P_n\right) \tag{26}$$

where $w_n$ is the weight assigned to the n-gram precision, typically uniform.

**METEOR**

METEOR is a recall-based metric that aligns words in the generated text with words in the reference text using synonyms, stemming, and paraphrasing. It combines precision, recall, and an alignment score to robustly capture linguistic variations. The METEOR score is calculated as follows:

$$METEOR = F_{mean} \cdot \left(1 - P_{frag}\right) \tag{27}$$

$$P_{frag} = 0.5 \left(\frac{chunks}{u_m}\right)^3 \tag{28}$$

where $P_{frag}$ is the fragmentation penalty, chunks denotes the number of matched chunks between the candidate and reference texts, and $u_m$ is the number of matched unigrams.

$$F_{mean} = \frac{10 \cdot P \cdot R}{R + 9P} \tag{29}$$

where $F_{mean}$ is the weighted harmonic mean of precision and recall.

$$P = \frac{u_m}{c_m} \tag{30}$$

where P is the unigram precision, and $c_m$ is the total number of unigrams in the candidate text.

$$R = \frac{u_m}{r_m} \tag{31}$$

where R is the unigram recall, with $r_m$ being the total number of unigrams in the reference text.

**GLEU**

GLEU, a variant of BLEU, addresses some of its shortcomings by considering both precision and recall of n-grams and is commonly used in evaluating machine translation. The GLEU score is computed as follows:

$$matches = \sum_{n\text{-}grams} \min\left(count_{candidate}(n\text{-}gram), count_{reference}(n\text{-}gram)\right) \tag{32}$$

Here, matches is the total count of matching n-grams between the candidate and reference texts, where $count_{candidate}(n-gram)$ and $count_{reference}(n-gram)$ represent the frequency of the n-gram in the candidate text and reference text, respectively. Next, the precision is computed as:

$$precision = \frac{matches}{\sum_{n\text{-}grams} count_{candidate}(n\text{-}gram)} \tag{33}$$

Similarly, the recall is computed as:

$$recall = \frac{matches}{\sum_{n\text{-}grams} count_{reference}(n\text{-}gram)} \tag{34}$$

Finally, the GLEU score is defined as:

$$GLEU = \min(precision, recall) \tag{35}$$

This final score represents the lower of the two values, capturing the balance between precision and recall.

## 4.5 Experimental Setup

In this section, we describe our training setup for both the PCI estimation network and the dense captioning network. Although these networks operate on the same dataset, they are trained separately due to their distinct objectives and architectures. After training, the outputs of both networks are combined to generate a comprehensive pavement condition assessment.

### PCI Estimation Network

For this network, all training images were resized to 416 × 416, preserving their aspect ratios by padding or cropping where necessary. The learning rate was fixed at 0.001, and the network was trained for 1,000 epochs. The primary loss function minimized during backpropagation was the MSE, although we also monitored the MAE to better understand error magnitude. Table 2 summarizes the key hyperparameters and training settings for the PCI estimation network. During training, we periodically evaluated the model on a validation split. We selected the best checkpoint based on the lowest MSE. Upon convergence, we obtained a model that accurately captures both coarse and fine details of the pavement surface (through YOLOv8 and SAM) and translates that information into a reliable PCI value via the CNN.

**Table 2.** Hyperparameters and training settings for the PCI estimation network

| **Hyperparameter** | **Value** |
|---|---|
| Image resolution | 416 × 416 |
| Number of epochs | 1,000 |
| Optimizer | Adam |
| Initial learning rate | 0.001 |
| Loss functions | MSE, MAE |
| Network components | YOLOv8, SAM, four-layer CNN |

### Dense Captioning Network

For the dense captioning network, we employed a YOLOv8 backbone that was frozen at the feature-extraction stage. The extracted feature maps were fed into a transformer encoder–decoder with a convolutional feed-forward module. We utilized two encoder layers and four decoder layers, each with eight heads. All embeddings (image and textual) had dimensions of 512. Training was carried out for 300 epochs, using early stopping (50 epochs patience) and a validation interval every two epochs. The Adam optimizer was also used, with both the encoder and the transformer set to a learning rate of 0.0001. Table 3 lists the main hyperparameters. Throughout training, cross-entropy served as the primary loss function for caption generation, and a doubly stochastic attention regularization term was added to encourage broad attention

coverage across the feature map. Once both networks were trained, each pavement image was processed independently by the PCI estimation network and the dense captioning network. The numerical PCI output and the generated dense caption were subsequently combined into the final output.

**Table 3.** Hyperparameters and training settings for the dense captioning network

| **Category** | **Parameter** | **Value** |
|---|---|---|
| Image encoder | Embedding dimension | 512 |
| Transformer | $d_{model}$ | 512 |
| Transformer | Encoder feed-forward dimension | 512 |
| Transformer | Decoder feed-forward dimension | 2,048 |
| Transformer | Number of encoder layers | 2 |
| Transformer | Number of decoder layers | 4 |
| Transformer | Number of encoder heads | 8 |
| Transformer | Number of decoder heads | 8 |
| Transformer | Dropout | 0.1 |
| Training protocol | Number of epochs | 300 |
| Training protocol | Validation interval | 2 |
| Training protocol | Early stopping patience | 50 |
| Training protocol | Learning rate patience | 5 |
| Training protocol | Transformer learning rate | 0.0001 |
| Training protocol | Encoder learning rate | 0.0001 |

# 5. RESULTS AND DISCUSSION

## 5.1 PCI Estimation Network

### Qualitative Results of the PCI Network

Our single-shot PCI estimation network demonstrated exceptional performance in predicting the exact PCI values for some of the images. This exceptional performance can be attributed to several key factors. First, the high precision of the YOLOv8 object detection accurately localizes pavement defects, such as cracks and potholes, ensuring that relevant features are captured effectively. The detailed segmentation masks generated by the SAM model further enhance this accuracy by providing precise boundaries of the defects. Additionally, the concatenation of the pavement images with these binary masks creates enriched feature representations, which the four-layer CNN leverages to learn robust and discriminative features. This dual representation of visual and segmented data significantly enhances the model's ability to generalize across diverse pavement conditions. The results have been illustrated in Figure 5.

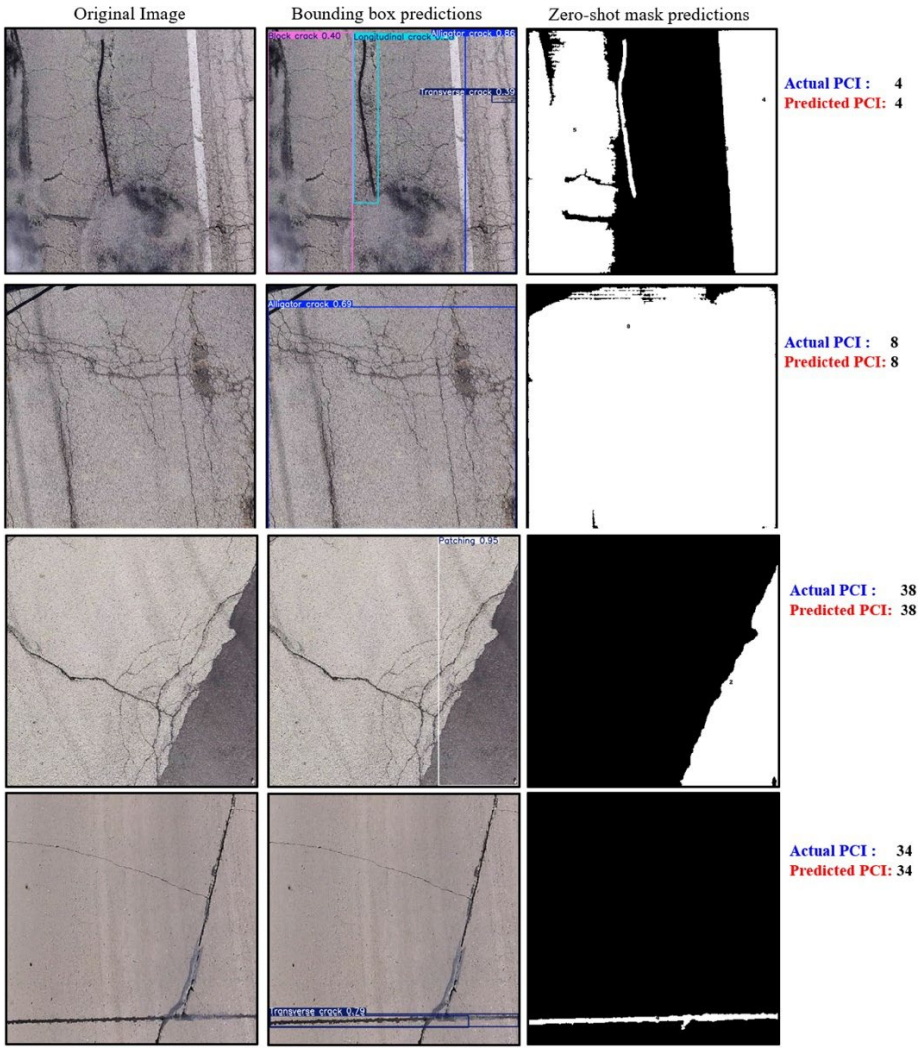



**Figure 5.** Exact PCI predictions for samples of images.

While our network generally performs exceptionally well, there are instances of good but not perfect predictions. For example, some predictions were slightly off, either marginally higher or lower than the actual PCI values, as shown in Figure 6. These minor discrepancies can be attributed to subtle defect patterns, which are challenging to detect and segment accurately; variations in lighting and shadowing conditions; fixed road elements such as drains; and the complexity of mixed defect types in some pavements. Additionally, the object detection model occasionally failed to recognize these challenging distresses, considering some of the images as background and generating entirely null masks. Despite these challenges, the model's predictions remain close to the actual PCI values, demonstrating its robustness.

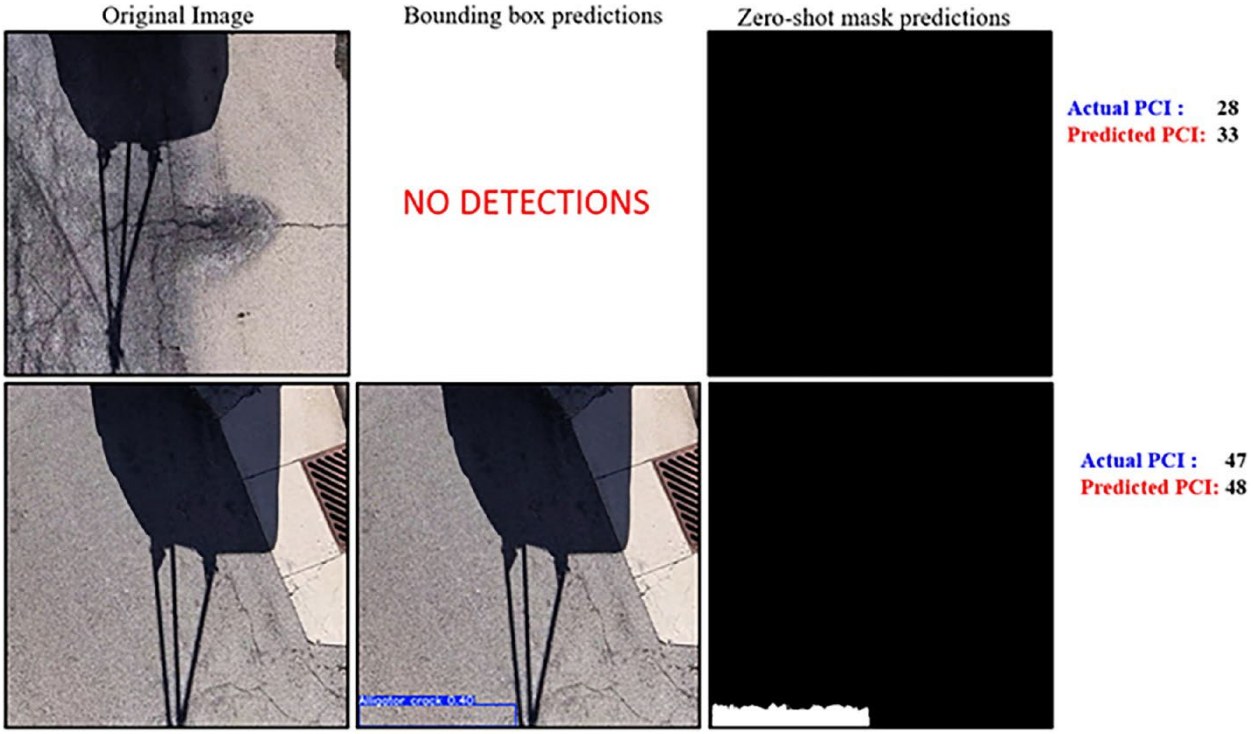



**Figure 6.** Instances of slightly off PCI predictions.

In some instances, our network exhibited poor performance, highlighting areas for potential improvement. Figure 7 shows some of those instances. A significant factor contributing to this is the quality of the segmentation masks generated by the SAM model. Some masks fail to accurately delineate the patterns in complex distresses such as alligator cracks and block cracks, often resulting in masks that cover the entire area of the bounding box rather than the specific distress. Additionally, the network occasionally produces incorrect detections, leading to erroneous masks that do not correspond to actual pavement defects. There are also cases where the model fails to detect all the distresses present in the image or misses detecting any distresses altogether, treating parts of the pavement as background. These issues in the detection and segmentation stages propagate through the network, affecting the subsequent feature extraction and learning processes of the CNN. When the input to the CNN includes inaccurately segmented masks or incorrect detections, the resulting feature representations are flawed, leading to significant deviations in the predicted PCI values.

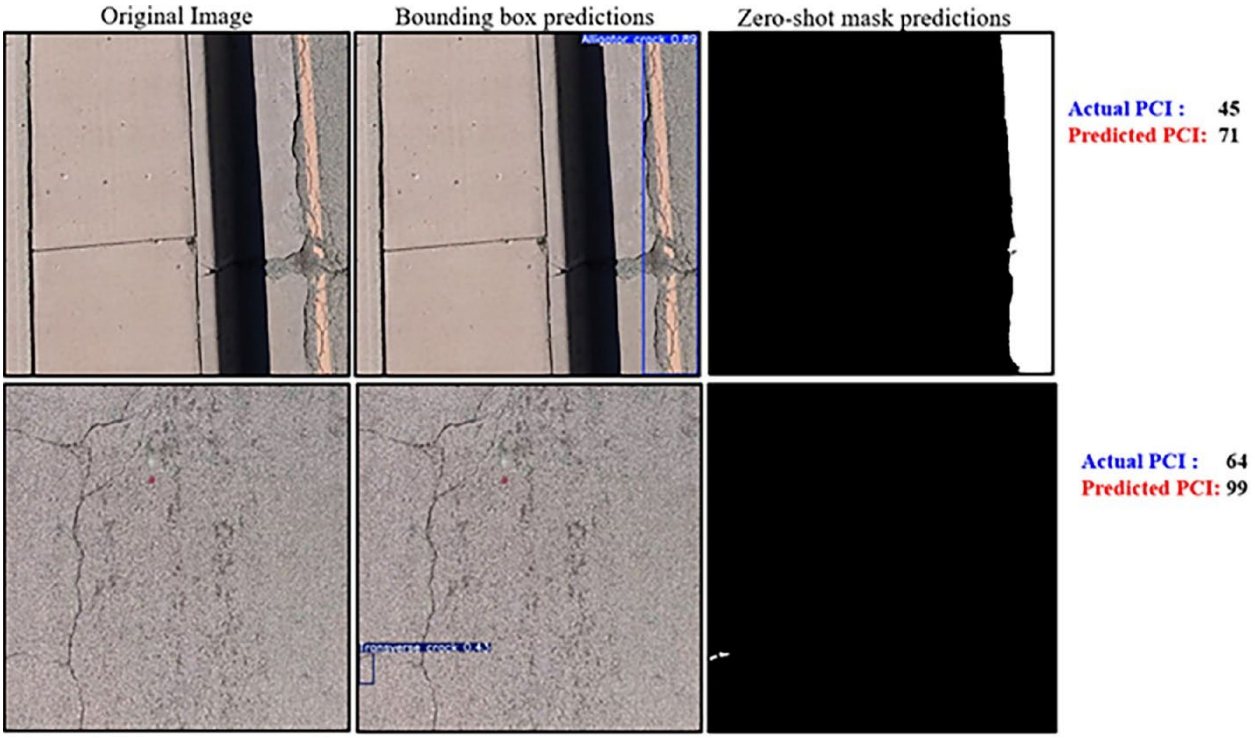



**Figure 7.** Instances of poor performance in PCI estimation.

## Error Analysis of the Network

To thoroughly evaluate the performance of our network, we conducted an error analysis focusing on the overprediction and underprediction of PCI values. Our analysis revealed that, on average, when the predicted PCI is higher than the actual PCI, the MAE is 37.79 with a standard deviation of 24.28. Conversely, when the predicted PCI is lower than the actual PCI, the MAE is 30.24 with a standard deviation of 25.24. Figure 8 illustrates the distributions of these errors. Although the model shows strong performance on average, a subset of predictions deviates from the actual PCI values. Reducing these large errors remains a key priority, and future work will focus on improved training strategies and architectural enhancements to address this gap. To understand the frequency of errors and identify the PCI ranges where the model tends to overpredict and underpredict, we plot a frequency distribution and a heatmap of the prediction errors, as illustrated in Figure 9.

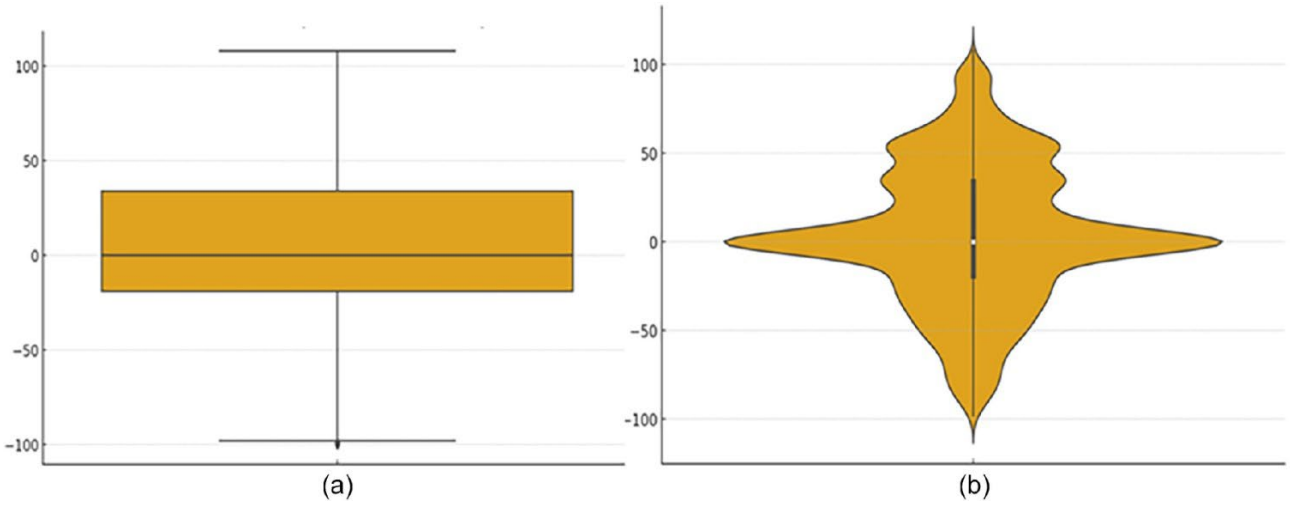



**Figure 8.** Error distribution in PCI estimation network.

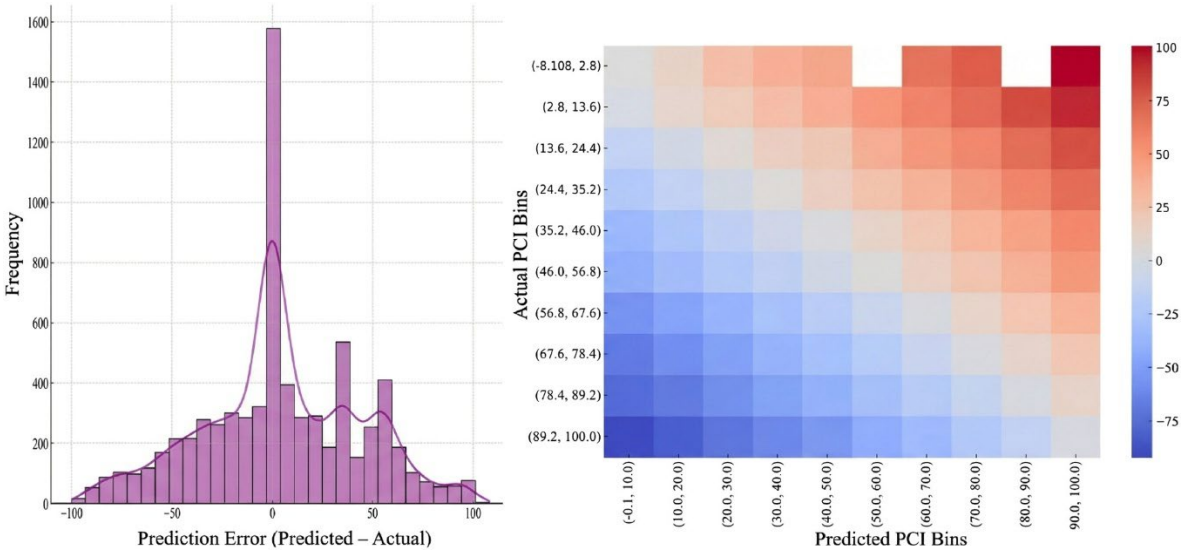



**Figure 9.** Frequency distribution and heatmap of prediction errors.

## 5.2 Dense Captioning Network

Our dense captioning network equally demonstrated great performance in generating detailed and accurate captions for pavement images. This performance resulted from the unique components of our dense captioning network such as the YOLOv8 backbone and the convolutional feed-forward module. The YOLOv8 backbone, trained specifically on pavement images, effectively extracts detailed hierarchical features, capturing both low-level textures and high-level semantic information relevant to pavement distresses. These rich feature maps serve as a robust input for the transformer encoder–decoder, which uses attention mechanisms to focus on relevant image areas. The convolutional feedforward layers further refine these features, enabling the model to generate accurate and contextually precise dense captions. This integration

allows the model to reliably describe various pavement conditions, as illustrated by the alignment of predicted captions with actual descriptions in Figure 10.

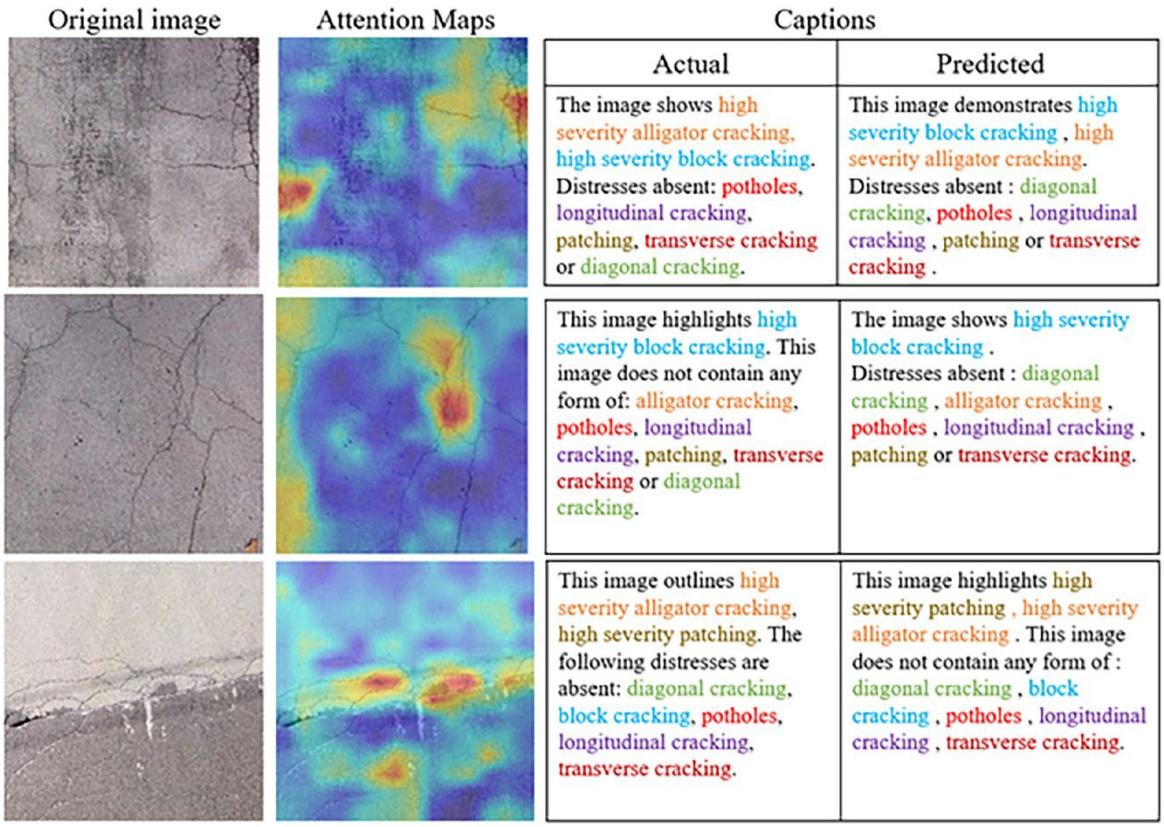



**Figure 10**. Examples of predicted accurate captions for sample pavement images.

However, there are instances where the network failed to generate accurate captions for the pavement images. For instance, in Figure 11(a), the network failed to include low severity patching as part of the distresses present in the image. This omission may be due to the model's reduced sensitivity to less prominent features, likely stemming from an imbalance in the training dataset where such features are underrepresented. In Figure 11(b), the network completely failed to identify medium severity longitudinal cracking, which suggests limitations in the model's ability to generalize to certain distress types.

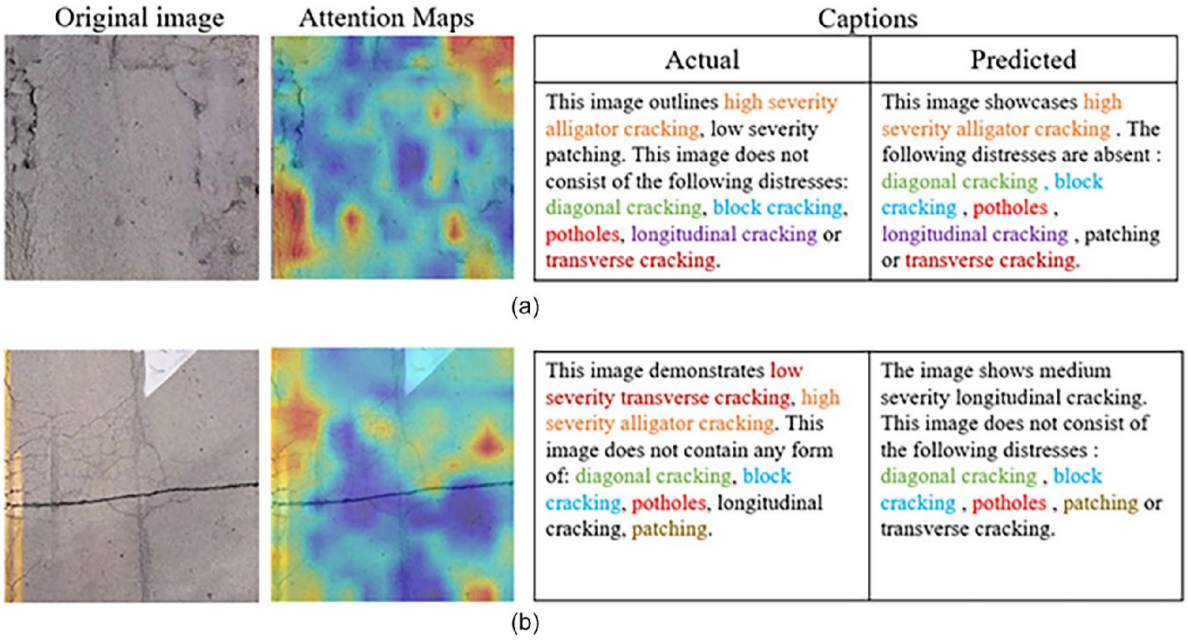



**Figure 11**. Instances of the dense captioning network's shortcomings.

An exceptional scenario is illustrated in Figure 12, where our network demonstrated its ability to correct an annotation error in the ground truth. The original ground truth annotation failed to include alligator cracking as part of the present distresses, while the model accurately generated a caption highlighting high severity alligator cracking, which was the correct distress type. This instance underscores the model's robustness to accurately identify pavement distresses even when the provided annotations are erroneous. This capability is particularly valuable in real-world applications, where annotation errors can be common, and the model's ability to self-correct contributes significantly to the overall accuracy and reliability of the inspection process. Figure 13 illustrates the overall framework for our dense captioning network.

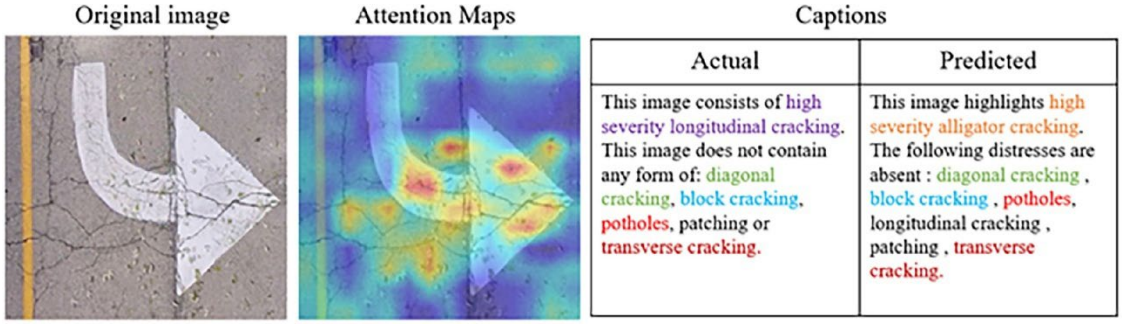



**Figure 12.** Model correcting annotation error.

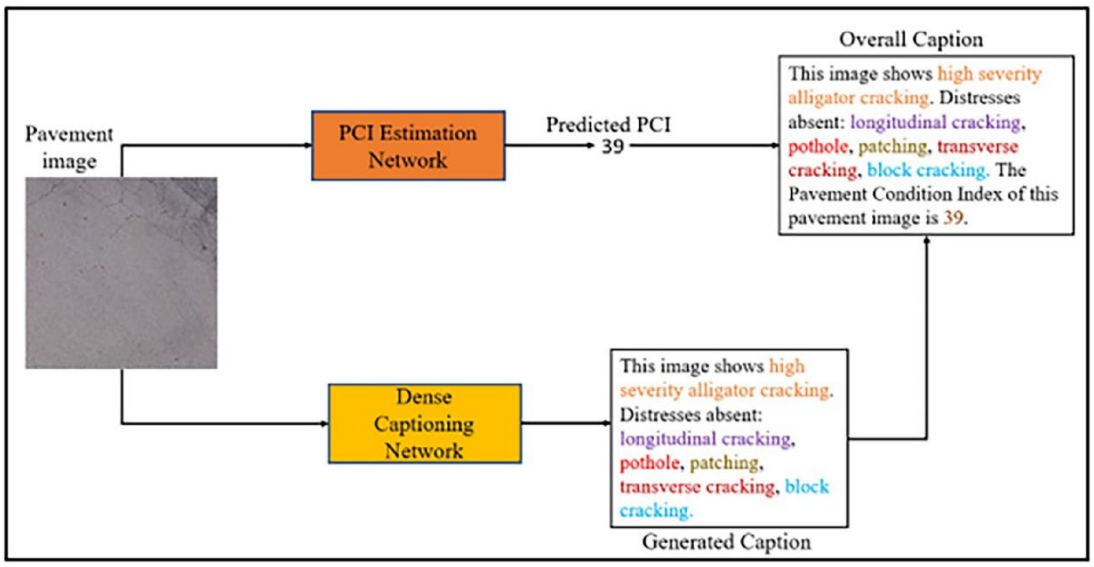



**Figure 13.** Overall framework for the dense captioning network (inference pipeline).

We evaluated the performance of our dense captioning model on our validation data using several standard metrics, including BLEU (1 to 4 n-grams), GLEU, and METEOR. Table 4 presents the mean and standard deviation for each metric based on our validation results.

**Table 4.** Mean and standard deviation of evaluation metrics for dense captioning

| Metric | Mean | Standard deviation |
|---|---|---|
| BLEU-1 | 0.2485 | ±0.1896 |
| BLEU-2 | 0.2201 | ±0.1694 |
| BLEU-3 | 0.1951 | ±0.1524 |
| BLEU-4 | 0.1756 | ±0.1391 |
| GLEU | 0.2442 | ±0.1413 |
| METEOR | 0.3211 | ±0.1677 |

Note. Mean values represent the average performance across generated captions, while standard deviations indicate the variability of the model's outputs across different captions.

## 5.3 Ablation Studies

### 5.3.1 PCI Estimation Network

To quantify the impact of the segmentation component on PCI estimation, we conducted an ablation study by removing the SAM from our PCI estimation network. In the full architecture, the YOLOv8-based object detector produces bounding boxes that serve as prompts for SAM, generating binary masks. These masks are concatenated with the original pavement image and then processed by a four-layer CNN to predict the PCI. By contrast, in the ablation experiment, the network is fed only the raw image, omitting the SAM-generated masks.

Experimental results reveal that when SAM is included, the model achieves a validation MSE of 474.13 and a validation MAE of 16.21, as shown in Table 5. Although the MSE may seem high relative to the 0 to 100 PCI scale, the results still demonstrate useful predictive performance.

However, we acknowledge that even small shifts across PCI category boundaries—such as from satisfactory to good—can impact maintenance decisions and associated cost estimates, particularly at the network level. In comparison, removing SAM leads to a slightly higher validation loss of 491.64 and a validation MAE of 16.43. This marginal degradation indicates that the segmentation masks generated by SAM provide complementary spatial cues, enabling the CNN to capture subtle pavement distress features more effectively. Thus, even though the performance difference is modest, these findings underscore the benefit of integrating segmentation information into the PCI estimation pipeline.

**Table 5**. Ablation study results for the PCI estimation network

| Configuration | Validation MSE | Validation MAE |
|---|---|---|
| With SAM | **474.13** | **16.21** |
| No SAM | 491.64 | 16.43 |

Note. Bold values indicate the best performance. Lower MSE and MAE values represent better prediction accuracy.

### 5.3.2 Dense Captioning Network

To investigate the impact of different backbone architectures and training strategies on the performance of our dense captioning network, we performed a comprehensive ablation study. Specifically, we evaluated six prominent object detection backbones—YOLOv8 (Jocher et al. 2023), YOLOv9 (Wang et al. 2024b), YOLOv10 (Wang et al. 2024a), YOLOv11 (Khanam and Hussain 2024), YOLOv12 (Tian et al. 2025), and RT-DETR (Lv et al. 2023)—under two distinct training scenarios: with the backbone parameters frozen and unfrozen. This analysis helps understand the importance of backbone choice and fine-tuning strategy in achieving optimal captioning performance.

Table 6 presents detailed performance metrics across all experimental setups. We observe consistently better performance across all models when the backbone weights are unfrozen during training. Notably, the YOLOv8 backbone achieves the highest BLEU and METEOR scores under unfrozen conditions (BLEU-4 = 0.2640, METEOR = 0.4253), highlighting the significant benefits of end-to-end training for this backbone. Conversely, the RT-DETR backbone yields the lowest metrics in both frozen and unfrozen settings, suggesting limited adaptability in the context of the dense captioning task.

Interestingly, while YOLOv9 exhibits a substantial performance improvement (BLEU-4 increasing from 0.1354 frozen to 0.2350 unfrozen), its absolute performance remains below that of YOLOv8 under identical conditions. YOLOv11 and YOLOv12 also demonstrate notable improvements with fine-tuning; however, these improvements are relatively modest compared with YOLOv8, suggesting diminishing returns in more recent YOLO variants for this particular application. As a result, YOLOv8 was selected as the main backbone for the dense captioning network based on its strong detection accuracy, efficient feature extraction, and superior performance in our ablation study. It consistently achieved the highest BLEU and METEOR scores across all tested configurations, demonstrating its effectiveness in capturing detailed pavement features critical for generating accurate captions.

**Table 6.** Performance metrics for different backbones used to train the dense captioning network

| Backbone | Frozen | BLEU-1 | BLEU-2 | BLEU-3 | BLEU-4 | GLEU | METEOR |
|---|---|---|---|---|---|---|---|
| YOLOv8 | Yes | 0.2485 | 0.2201 | 0.1951 | 0.1756 | 0.2442 | 0.3211 |
| YOLOv8 | No | **0.3799** | **0.3317** | **0.2930** | **0.2640** | **0.3234** | **0.4253** |
| YOLOv9 | Yes | 0.1924 | 0.1691 | 0.1494 | 0.1354 | 0.1982 | 0.2735 |
| YOLOv9 | No | 0.3501 | 0.3095 | 0.2669 | 0.2350 | 0.3011 | 0.4109 |
| YOLOv10 | Yes | 0.2717 | 0.2418 | 0.2123 | 0.1901 | 0.2525 | 0.3283 |
| YOLOv10 | No | 0.3133 | 0.2748 | 0.2364 | 0.2082 | 0.2806 | 0.3762 |
| YOLOv11 | Yes | 0.2904 | 0.2515 | 0.2190 | 0.1940 | 0.2641 | 0.3538 |
| YOLOv11 | No | 0.3320 | 0.2913 | 0.2537 | 0.2244 | 0.3014 | 0.4121 |
| YOLOv12 | Yes | 0.2789 | 0.2445 | 0.2129 | 0.1876 | 0.2603 | 0.3466 |
| YOLOv12 | No | 0.3113 | 0.2748 | 0.2391 | 0.2103 | 0.2926 | 0.3946 |
| RT-DETR | Yes | 0.1809 | 0.1594 | 0.1428 | 0.1306 | 0.1795 | 0.2440 |
| RT-DETR | No | 0.2134 | 0.1875 | 0.1617 | 0.1415 | 0.2102 | 0.2929 |

Note. Bold values indicate the best performance for each metric.

These results clearly indicate the critical role of backbone fine-tuning in optimizing dense captioning performance. They underscore that carefully choosing and fine-tuning the backbone significantly impacts the overall effectiveness of captioning models, with earlier YOLO variants (particularly YOLOv8) displaying superior adaptability and performance in our experiments.

# 6. Conclusions and Recommendations

This study addressed the challenge of automating pavement condition assessment by developing a multimodal framework that combines computer vision and natural language processing. The proposed approach consisted of two main components: a PCI estimation network and a dense captioning network. Together, these components integrated visual pavement information with natural language descriptions to support both quantitative condition assessment and descriptive interpretation.

The PCI estimation network achieved a positive correlation between predicted and actual PCI values, demonstrating its potential for automating pavement condition assessment. The dense captioning network also generated textual outputs that described the condition of pavement images. Notably, the model showed robustness in handling complex scenarios and, in some cases, appeared to correct annotation inconsistencies in the ground-truth data.

This study focused on distress types that are visually prominent and frequently encountered in roadway pavements. However, the current framework does not address several other distress types, such as rutting, bleeding, or shoving. Expanding the dataset to include these additional distress categories would be necessary to improve the generalizability of the model for broader pavement and airfield applications. In addition, the performance of the proposed framework is influenced by the quality and consistency of the input data. In real-world settings, image noise, lighting variability, and incomplete annotations can introduce additional challenges.

For future work, we recommend exploring advanced vision-language models for pavement assessment tasks. This would require the creation of a diverse multimodal dataset that combines pavement images with corresponding textual descriptions. With such an enhanced dataset, future research could expand beyond image captioning to address more complex tasks such as visual question answering, visual reasoning, and visual grounding. Incorporating these advanced capabilities could improve the interpretability and versatility of automated pavement assessment systems.

Ultimately, these advancements have the potential to support infrastructure management and decision-making in pavement maintenance. Future work will also explore the integration of attention mechanisms and denoising modules to improve segmentation robustness and enhance PCI estimation accuracy, particularly under challenging pavement conditions.

# 7. REFERENCES


1. Agyei Kyem, B., J. K. Asamoah, and A. Aboah. 2025. "Context-CrackNet: A context-aware framework for precise segmentation of tiny cracks in pavement images." *Constr. Build. Mater*. 484 (7): 141583. https://doi.org/10.1016/j.conbuildmat.2025.141583.

2. Agyei Kyem, B., J. K. Asamoah, E. Denteh, A. Danyo, and A. Aboah. 2026. "Self-supervised multi-scale transformer with attention-guided fusion for efficient crack detection." *Autom. Constr*. 181 (Jan): 106591. https://doi.org/10.1016/j.autcon.2025.106591.

3. ASTM. 2024. *Standard Practice for Roads and Parking Lots Pavement Condition Index Surveys.* West Conshohocken, PA: ASTM.

4. Biradar, V. G., G. Mukund, S. Agarwal, S. K. Singh, and R. U. Bharadwaj. 2023. "Leveraging deep learning model for image caption generation for scenes description." In *Proc., 2023 Int. Conf. on Evolutionary Algorithms and Soft Computing Techniques*, 1–5. New York: IEEE.

5. Chavan, A., S. Pimplikar, and A. Deshmukh. 2022. "An overview of machine learning techniques for evaluation of pavement condition." In *Proc., 2022 IEEE 4th Int. Conf. on Cybernetics, Cognition and Machine Learning Applications*, 139–143. New York: IEEE.

6. Danyo, A., A. Dontoh, and A. Aboah. 2025. "An Improved ResNet50 Model for Predicting Pavement Condition Index (PCI) Directly from Pavement Images." *Road Mater. Pavement Des.* 1–18. https://doi.org/10.1080/14680629.2025.2498632.

7. Du, Y., N. Pan, Z. Xu, F. Deng, Y. Shen, and H. Kang. 2021. "Pavement distress detection and classification based on YOLO network." *Int. J. Pavement Eng*. 22 (1): 1659–1672. https://doi.org/10.1080/10298436.2020.1714047.

8. Gyamfi, Y. A., B. Buttlar, E. Dave, D. Mensching, and H. Majidifard. 2024. "DSPS 24 student competition." Accessed December 12, 2024. https://dsps-1e998.web.app/.

9. Han, C., T. Ma, J. Huyan, X. Huang, and Y. Zhang. 2022. "CrackW-Net: A Novel Pavement Crack Image Segmentation Convolutional Neural Network." *IEEE Trans. Intell. Transp. Syst.* 23 (11): 22135–22144. https://doi.org/10.1109/TITS.2021.3095507.

10. Han, S., I.-H. Chung, Y. Jiang, and B. Uwakweh. 2023. "PCIer: Pavement condition evaluation using aerial imagery and deep learning." *Geographies 3* (1): 132–142. https://doi.org/10.3390/geographies3010008.

11. Hoang, N.-D., and Q.-L. Nguyen. 2019. "A novel method for asphalt pavement crack classification based on image processing and machine learning." *Eng. Comput*. 35 (5): 487–498. https://doi.org/10.1007/s00366-018-0611-9.

12. Ibragimov, E., Y. Kim, J. H. Lee, J. Cho, and J.-J. Lee. 2024. "Automated pavement condition index assessment with deep learning and image analysis: An end-to-end approach." *Sensors 24* (7): 2333. https://doi.org/10.3390/s24072333.

13. Jocher, G., J. Qiu, and A. Chaurasia. 2023. “Ultralytics YOLOv8.” Accessed December 9, 2024. https://ultralytics.com.

14. Kandala, H., S. Saha, B. Banerjee, and X. X. Zhu. 2022. “Exploring transformer and multilabel classification for remote sensing image captioning.” *IEEE Geosci. Remote Sens. Lett.* 19 (Aug): 1–5. https://doi.org/10.1109/LGRS.2022.3198234.

15. Karballaeezadeh, N., S. D. Mohammadzadeh, D. Moazemi, S. S. Band, A. Mosavi, and U. Reuter. 2020. “Smart structural health monitoring of flexible pavements using machine learning methods.” *Coatings 10* (11): 1100. https://doi.org/10.3390/coatings10111100.

16. Karim, F. M., K. A. H. Rubasi, and A. A. Saleh. 2016. “The road pavement condition index (PCI) evaluation and maintenance: A case study of Yemen.” *Organ. Technol. Manage. Constr. Int. J.* 8 (1): 1446–1455. https://doi.org/10.1515/otmcj-2016-0008.

17. Karthik, A. S., M. H. K. Karthik, and S. Yashwanth. 2024. “Image captioning: Analyzing CNN-LSTM and vision-GPT models.” In *Proc., 2024 IEEE 9th Int. Conf. for Convergence in Technology,* 1–6. New York: IEEE.

18. Khanam, R., and M. Hussain. 2024. “YOLOv11: An Overview of the Key Architectural Enhancements.” Preprint, submitted October 23, 2024. https://arxiv.org/abs/2410.17725.

19. Kushal, K., M. Manoj, K. Reddy, and P. C. Nair. 2024. “An in-depth exploration of image captioning training approaches and performance analysis.” In *Proc., 2024 IEEE 9th Int. Conf. for Convergence in Technology*, 1–7. New York: IEEE.

20. Kyem, B. A., E. K. O. Denteh, J. K. Asamoah, K. A. Tutu, and A. Aboah. 2024. “Advancing pavement distress detection in developing countries: A novel deep learning approach with locally-collected datasets.” Preprint, submitted August 10, 2024. https://arxiv.org/abs/2408.05649.

21. Kyem, B. A., N. J. Owor, A. Danyo, J. K. Asamoah, E. Denteh, T. Muturi, A. Dontoh, Y. Adu-Gyamfi, and A. Aboah. 2025. “Task-specific dual-model framework for comprehensive traffic safety video description and analysis.” In *Proc., IEEE/CVF Int. Conf. on Computer Vision*, 5325–5333. New York: IEEE.

22. Lala, M. P., and D. Kumar. 2022. “Improving the quality of image captioning using CNN and LSTM method.” In *Proc., 2022 Int. Conf. on Intelligent Innovations in Engineering and Technology*, 64–70. New York: IEEE.

23. Liu, W., S. Chen, L. Guo, X. Zhu, and J. Liu. 2021. “CPTR: Full Transformer Network for Image Captioning.” Preprint, submitted January 28, 2021. https://arxiv.org/abs/2101.10804.

24. Lv, W., S. Xu, Y. Zhao, G. Wang, J. Wei, C. Cui, Y. Du, Q. Dang, and Y. Liu. 2023. “DETRs Beat YOLOs on Real-time Object Detection.” New York: IEEE.

25. Majidifard, H., Y. Adu-Gyamfi, and W. G. Buttlar. 2020a. “Deep machine learning approach to develop a new asphalt pavement condition index.” *Constr. Build. Mater.* 247 (6): 118513. https://doi.org/10.1016/j.conbuildmat.2020.118513.

26. Majidifard, H., P. Jin, Y. Adu-Gyamfi, and W. G. Buttlar. 2020b. “Pavement image datasets: A new benchmark dataset to classify and densify pavement distresses.” *Transp. Res. Rec.* 2674 (1): 328–339. https://doi.org/10.1177/0361198120907283.

27. Mandal, V., L. Uong, and Y. Adu-Gyamfi. 2018. “Automated road crack detection using deep convolutional neural networks.” In *Proc., 2018 IEEE Int. Conf. on Big Data*, 5212–5215. New York: IEEE.

28. Mishra, S. K., R. Dhir, S. Saha, P. Bhattacharyya, and A. K. Singh. 2021. “Image captioning in Hindi language using transformer networks.” *Comput. Electr. Eng.* 92 (6): 107114. https://doi.org/10.1016/j.compeleceng.2021.107114.

29. Nie, M., and K. Wang. 2018. “Pavement distress detection based on transfer learning.” In *Proc., 2018 5th Int. Conf. on Systems and Informatics*, 435–439. New York: IEEE.

30. Osman, M. K., M. E. Zamree, M. Idris, K. A. Ahmad, N. A. Yusof, A. Ibrahim, A. H. Rabiain, and I. Bahri. 2021. “Pavement crack classification using deep convolutional neural network.” *J. Mech. Eng.* 10 (1): 227–244.

31. Owor, N. J., H. Du, A. Daud, A. Aboah, and Y. Adu-Gyamfi. 2023. “Image2PCI—A Multitask Learning Framework for Estimating Pavement Condition Indices Directly from Images.” Preprint, submitted October 12, 2023. https://arxiv.org/abs/2310.08538.

32. Prajwal Gowda, M., M. Rohini, and K. P. Bindu. 2024. “Captionify: Bridging the Gap Between Vision and Language with Neural Networks.” *Int. J. Res. Appl. Sci. Eng. Technol.* 12 (1): 2155–2160. https://doi.org/10.22214/ijraset.2024.61513.

33. Qureshi, W. S., S. I. Hassan, S. McKeever, D. Power, B. Mulry, K. Feighan, and D. O’Sullivan. 2022. “An exploration of recent intelligent image analysis techniques for visual pavement surface condition assessment.” *Sensors* 22 (22): 9019. https://doi.org/10.3390/s22229019.

34. Shao, Z., J. Han, D. Marnerides, and K. Debattista. 2022. “Region-object relation-aware dense captioning via transformer.” *IEEE Trans. Neural Networks Learn. Syst.* 36 (3): 4184–4195. https://doi.org/10.1109/TNNLS.2022.3152990.

35. Sirhan, M., S. Bekhor, and A. Sidess. 2022. “Implementation of deep neural networks for pavement condition index prediction.” *J. Transp. Eng., Part B: Pavements* 148 (1): 04021070. https://doi.org/10.1061/JPEODX.0000333.

36. Stricker, R., D. Aganian, M. Sesselmann, D. Seichter, M. Engelhardt, R. Spielhofer, M. Hahn, A. Hautz, K. Debes, and H.-M. Gross. 2021. “Road surface segmentation—Pixel-perfect distress and object detection for road assessment.” In *Proc., 2021 IEEE 17th Int. Conf. on Automation Science and Engineering*, 1789–1796. New York: IEEE.

37. Tian, Y., Q. Ye, and D. Doermann. 2025. “YOLOv12: Attention-centric real-time object detectors.” Preprint, submitted February 18, 2025. https://arxiv.org/abs/2502.12524.

38. Tong, Z., D. Yuan, J. Gao, Y. Wei, and H. Dou. 2020. “Pavement-distress detection using ground-penetrating radar and network in networks.” *Constr. Build. Mater.* 233 (Feb): 117352. https://doi.org/10.1016/j.conbuildmat.2019.117352.

39. Wang, A., H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, and G. Ding. 2024a. “YOLOv10: Real-time end-to-end object detection.” Preprint, submitted October 30, 2024. https://arxiv.org/abs/2405.14458.

40. Wang, C., I. Yeh, and H. Liao. 2024b. “YOLOv9: Learning what you want to learn using programmable gradient information.” Preprint, submitted February 29, 2024. https://arxiv.org/abs/2402.13616.

41. Wang, D., H. Hu, and D. Chen. 2020. “Transformer with sparse self-attention mechanism for image captioning.” *Electron. Lett.* 56 (15): 764–766. https://doi.org/10.1049/el.2020.0635.

42. Yan, C., Y. Hao, L. Li, J. Yin, A. Liu, Z. Mao, Z. Chen, and X. Gao. 2022. “Task-adaptive attention for image captioning.” *IEEE Trans. Circuits Syst. Video Technol.* 32 (1): 43–51. https://doi.org/10.1109/TCSVT.2021.3067449.

43. Yi, C., J. Liu, T. Huang, H. Xiao, and H. Guan. 2023. “An efficient method of pavement distress detection based on improved YOLOv7.” *Meas. Sci. Technol.* 34 (11): 115402. https://doi.org/10.1088/1361-6501/ace929.

44. Yu, J., J. Li, Z. Yu, and Q. Huang. 2020. “Multimodal transformer with multi-view visual representation for image captioning.” *IEEE Trans. Circuits Syst. Video Technol.* 30 (12): 4467–4480. https://doi.org/10.1109/TCSVT.2019.2947482.

45. Zafar, M. S., S. N. R. Shah, M. J. Memon, T. A. Rind, and M. A. Soomro. 2019. “Condition survey for evaluation of pavement condition index of a highway.” *Civ. Eng. J.* 5 (6): 1367–1383. https://doi.org/10.28991/cej-2019-03091338.

46. Zakeri, H., F. M. Nejad, and A. Fahimifar. 2017. “Image based techniques for crack detection, classification and quantification in asphalt pavement: A review.” *Arch. Comput. Methods Eng.* 24 (11): 935–977. https://doi.org/10.1007/s11831-016-9194-z




NDSU Dept 2880 • P.O. Box 6050
Fargo, ND 58108-6050

(701)231-7767

ndsu.ugpti@ndsu.edu