MoPac+: Anchor Free **Mo**deling of the **Pa**vement **C**racks using Hard-Example Mining and Weighted Loss⋆

Debojyoti Biswas[a,1], Andrew Scouten[a,1], Haitao Gong[a,2], Feng Wang[a,2] and Jelena Tešić[a,*,1]

[a]*Texas State University, 601 University Ave, San Marcos 78666, Texas, U.S.A.*

ARTICLE INFO

Keywords:
deep learning; pavement crack detection; roadway safety; anchor-free detection;

ABSTRACT

Automated pavement distress (crack) detection through image analysis remains critical for pavement condition evaluation, maintenance planning, project selection, and asset management. While deep learning has significantly advanced pavement distress detection, current datasets and methodologies predominantly focus on common crack types (e.g., longitudinal/transverse), leaving rare distress patterns and fine-grained categorizations inadequately addressed. Existing approaches further demonstrate limitations in handling irregular crack morphologies and data-scarce scenarios. This paper introduces *MoPac+*, a novel anchor-free framework integrating multi-stage crack modeling with adaptive hard-example mining and loss reweighting, specifically designed for robust pavement deterioration localization and classification from pavement intensity imagery. Our methodology addresses two key challenges: (1) uncontrolled crack geometry with significant intra-class size variation, addressed through anchor-free detection modules, and (2) grayscale image limitations mitigated by pre-processing techniques that enhance distressed regions through strategic dilation and class-balancing operations. The proposed system particularly improves the detection efficiency of rare classes by implementing gradient-harmonized loss allocation and hierarchical feature fusion. Comprehensive experiments demonstrate substantial performance gains, achieving a **6.9%** relative improvement in mean average precision (mAP) over baseline YOLOv8 models while maintaining computational efficiency. These advancements establish new state-of-the-art benchmarks for imbalanced pavement distress recognition in resource-constrained environments.

1. Introduction

Efficient pavement management is fundamental to maintaining highway infrastructure in good condition and ensuring safety to the driving public, as pavement structures are continually exposed to environmental weathering and heavy traffic loads that accelerate deterioration. Early and accurate detection of pavement cracks and other forms of distress is vital for extending service life and optimizing maintenance and rehabilitation (M&R) activities. The Texas Department of Transportation (2023) maintains an extensive roadway network spanning over 703 thousand lane miles and supporting an annual travel volume of nearly 301 billion miles. Simple arithmetic results in the fact that each average lane-mile interacts with vehicles over 428 thousand times a year, and such a scale highlights the necessity for robust automated pavement crack detection systems capable of facilitating timely and cost-effective maintenance. Automated pavement crack detection typically relies on advanced imaging sensors and computer vision algorithms to identify and characterize surface defects. However, accurate detection remains challenging due to the inherent complexity and variability of pavement distress types, materials, pavement types, and surface texture characteristics. Pavement surfaces may exhibit a wide spectrum of defects, from minor hairline cracks to severe structural failures, with each pavement type, such as asphalt or concrete, presenting unique visual characteristics. Sensor heterogeneity, variations in image quality, intra-class diversity, and inter-class similarities further complicate the detection process, often limiting the generalizability and reliability of automated solutions.

⋆ This research has been partially funded by the TxDOT Project #0-7150 and NSF Project 2213694

*Corresponding author

ubq3@txstate.edu (D. Biswas); yzb2@txstate.edu (A. Scouten); h_g153@txstate.edu (H. Gong); f_w34@txstate.edu (F. Wang); jtesic@txstate.edu (J. Tešić)

ORCID(s): 0000-0002-8842-0207 (D. Biswas); 0000-0001-0000-0000 (A. Scouten); 0000-0001-5818-8411 (H. Gong); 0000-0002-1528-9711 (F. Wang); 0000-0002-9972-9760 (J. Tešić)

[1]Computer Science

[2]Civil Engineering

Recent advancements in deep learning, particularly convolutional neural networks (CNNs), have substantially improved the accuracy of pavement crack detection and classification, for example, Chen, Liu, Chen, Zhu, Zhang and Wang (2024); Saberironaghi and Ren (2024); Zhou, Yang, Zhang, Zhang, Qu, Punetha, Li and Li (2025). State-of-the-art object detectors, including single-stage models like YOLO by Redmon, Divvala, Girshick and Farhadi (2016) and SSD by Liu, Anguelov, Erhan, Szegedy, Reed, Fu and Berg (2016), as well as two-stage frameworks such as Faster R-CNN and Mask R-CNN by Ren, He, Girshick and Sun (2015), have demonstrated strong performance on well-defined and prominent pavement damage. Despite these advances, existing approaches often struggle with detecting small, irregular, or rare distress types and typically provide only coarse localization. The challenges are exacerbated by the scarcity of comprehensive, annotated datasets and the lack of standardized benchmarks, which hinder the development, evaluation, and comparison of new methods, as evidenced in the research by Cao, Yang and Yang (2020); Tran, Tran, Lee, Kim, Baek and Nguyen (2021); Zheng, Xiao, Wang, Wu, Chen, Yuan and Jiang (2024); Deng, Yuan, Long, Chun, Chen and Chu (2024); Gong, Tešić, Tao, Luo and Wang (2023).

The main contributions of this paper are: (i) the development of advanced data augmentation and enhancement strategies to improve dataset quality and model generalizability; (ii) introduction of an efficient anchor-free crack detection architecture; (iii) implementation of hard-example mining using a customized focal loss function; and (iv) incorporation of weight-balancing mechanisms tailored to the unique characteristics of pavement distress datasets. Specifically, we propose *MoPac+*, **Mo**deling **Pa**vement **c**racks, a novel anchor-free detection framework designed to robustly localize and classify both common and rare pavement distress classes, even in data-constrained settings. As depicted in Figure 1, *MoPac+* integrates a multi-stage feature extraction backbone, a neck for multi-scale feature aggregation, and an anchor-free detection head for precise bounding box regression and classification. The anchor-free architecture is particularly suited to the diverse and uncontrolled geometries of pavement cracks, overcoming the limitations of traditional anchor-based detectors. To further enhance detection performance, especially for rare and subtle distress types, we introduce targeted data augmentation, image enhancement, and a loss reweighting strategy that incorporates focal loss and weight balancing to address class imbalance and prioritize hard examples. The remainder of this paper is organized as follows: Section 2 reviews related work; Section 3 details the proposed *MoPac+* methodology; Section 4 describes the dataset creation and characteristics; Section 5 presents experimental results and analysis; and Section 6 concludes with a summary of findings and future research directions.

2. Related Work

Liu, Xu, Yang, Niu and Pan (2008) pioneered early pavement crack detection methods that operated under the assumption that crack pixels are darker than their surroundings, and introduced thresholding algorithms, such as the adjacent difference histogram approach, that effectively distinguished crack regions from non-crack areas by maximizing pixel contrast. Liu et al. (2008) further advanced detection through a connected domain algorithm with directional segmentation, which leveraged spatial relationships for improved accuracy. Despite their computational efficiency, these threshold-based techniques are highly sensitive to image noise and often yield false positives due to their reliance on grayscale intensity alone. Next, Gavilán, Balcones, Marcos, Llorca, Sotelo, Parra, Ocaña, Aliseda, Yarza and Amírola (2011) reduced false positives by analyzing gray values along object contours, while Li and Liu (2008) segmented images into overlapping sub-regions to better detect small-scale and complex cracks. However, these traditional methods struggle with large-scale distress and cluttered backgrounds.

Redmon et al. (2016) marked a significant shift in the field with the introduction of the YOLO (You Only Look Once) family of models, which enabled real-time crack detection by predicting bounding boxes and class labels directly from input images. Successive iterations, notably YOLOv5, have incorporated multi-scale prediction, architectural enhancements, and advanced data augmentation, establishing them as leading baselines for pavement analysis. Recent studies, such as those by Hu, Yang, Jin and Fan (2023), Yu and Zhou (2023), and Roy and Bhaduri (2023), have further refined YOLO-based models by integrating transformer modules to capture long-range dependencies and contextual information. Wu, Wu and Xie (2024) proposed a lightweight crossed-feature pyramid network and an improved loss function, achieving a mean average precision (mAP) of 69.3% on challenging datasets. These advancements have enabled robust detection under varying conditions, including shadows and occlusions. The widespread adoption of YOLOv5 in works by Ye, Qu, Tao, Dai, Mao and Jin (2023), Pham, Nguyen and Donan (2022), and Hu, Hu, Yang, Huang and Li (2021) underscores its effectiveness for high-precision, real-time pavement crack analysis. On the other hand, Zhu, Su, Lu, Li, Wang and Dai (2020) proposes a Deformable Transformer for lightweight, end-to-end object

detection, focusing on small objects. However, most research has focused on common crack types, such as longitudinal and transverse cracks, leading to class imbalance and limited generalizability.

He, Gkioxari, Dollár and Girshick (2017) and Cai and Vasconcelos (2019) introduced two-stage detectors, such as Mask R-CNN and Cascade R-CNN, for pavement distress detection. These models utilize region proposal networks and multi-stage refinement to enhance detection accuracy, particularly in complex scenes. Comparative analyses by Son Dong Nguyen, Tran, Tran, Lee, Piran and Le (2022) have demonstrated that Faster R-CNN outperforms other deep learning methods for crack detection on high-resolution datasets. Further innovations include multi-layer feature fusion networks (Li, Xie, Gong, Yu, Xu, Sun and Wang (2021)), sensitivity detection networks for complex backgrounds (Huyan, Li, Tighe, Zhai, Xu and Chen (2019)), and transformer-based segmentation for real-time pixel-level analysis (Kang and Cha (2022)). While these approaches achieve high accuracy on standard datasets, their performance often degrades when detecting small or rare distress types or when applied to non-standard image resolutions.

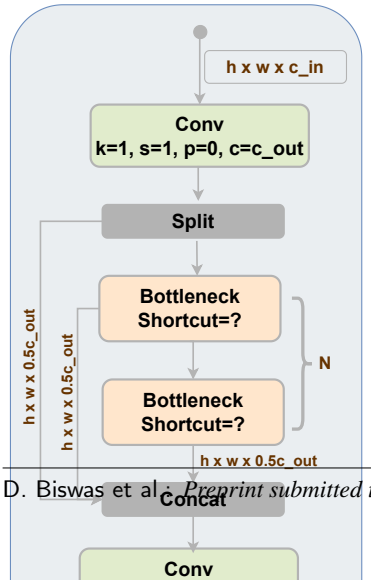
Although significant progress has been made, both single-stage and two-stage detection approaches still face challenges when applied to pavement crack detection. Most of this existing research primarily includes standard recurrent distress classes, such as longitudinal and transverse cracks, and the rare classes or more detailed and specific categorizations are often neglected. This imbalance leads to models proficient at identifying common distresses but cannot accurately detect rarer and critical pavement issues, limiting their practical application. Hence, the model improvements remain essential for supporting the durability and upkeep of transportation infrastructure worldwide.

3. MoPac+ AI Methodology

The growing importance of AI in pavement condition assessment necessitates the development of a practical yet robust Deep Neural Network (DNN) based distress detection model that can handle real-world data. Earlier, Scouten, Gong, Tešić and Wang (2025) introduced the *MoPac* skeleton baseline for practical deployment in road maintenance and safety assessment to address the class imbalance in the dataset that biases the detection process. To do this, *MoPac* utilized YOLOv5 and YOLOv8 as feature extractors while experimenting with different SVM and multi-scale detection heads. The experiments from *MoPac* showed high AP in recurrent labels (AP=0.96 for Joint Cracking, with 10,283 annotations) and very low AP in rare labels (AP=0.02 for Failed Concrete Patching, with 27 annotations) using an unmodified YOLOv8. The *MoPac* implementation achieved very marginal improvements for rare labels at the cost of detection accuracy of recurrent labels. At the same time, the unmodified YOLOv8 experiments show that the recurrent labels are overtaking the rare labels during training. *MoPac* 's experiments highlight the need for more robust methods, data collection, and annotation validation.

In this work, we propose the *MoPac+* architecture, an improvement over the baseline *MoPac* implementation. *MoPac+* still leverages the You Only Look Once (YOLO) deep learning algorithm while also introducing specific enhancements tailored to the distinctive characteristics of pavement imagery classes. The overall architecture of YOLOv8 Jocher and the Ultralytics team (2023) is presented in Figure 1, which shows that the YOLOv8 model does not have an explicit Region Proposal Network (RPN).

The model is divided into three modules: the Backbone (for deep Feature Extraction), the Neck (FPN + PAN using C2f, Upsample, Concat), and the Detection Head for classification.



YOLOv8 utilizes various reusable blocks or modules, which are combinations of DNN layers. Each *Convolutional* (*Conv.*) block utilizes a Conv2d layer with a 3×3 kernel, a stride of two, and a padding of one, followed by a BatchNorm2d layer and a Sigmoid Linear Unit (SiLU) activation function. The *CSP Bottleneck with two convolutions, faster* (*C2f*) block, as seen in Figure 3, builds upon the principles of CSP connections, which enable efficient gradient flow, reduce computational redundancy, and enhance feature reuse. Within the *C2f* block exists *Bottleneck* modules, which consist of two sequential *Conv.* blocks—where, if "Shortcut" is specified, the input vector to the *Bottleneck* module is added to the output vector of the final *Conv.* block in the module. The block's architectural design helps to preserve pixel-level details in deeper layers, which is particularly important for accurately detecting and classifying small objects within the dataset Yaseen (2024); Jocher and the Ultralytics team (2023).

The Deep Feature Extractor in YOLOv8 employs a *CSP-DarkNet53* Bochkovskiy, Wang and Liao (2020) network as the Backbone due to its

demonstrated efficacy in detecting small objects Biswas and Tešić (2022) and preserving spatial information across deep convolutional layers Biswas and Tešić (2024);Yaseen (2024). Figure 2 illustrates the proposed architecture, combining sequential *Conv.* and *C2f* modules across the backbone to help guarantee stability in feature extraction and enable robust multi-scale representation learning Jocher and the Ultralytics team (2023). Finally, the *Spatial Pyramid Pooling - Fast* (*SPPF*) module builds upon the traditional Spatial Pyramid Pooling (SPP) structure, refining the concept by applying a series of fixed-size max pooling operations— typically using a 5×5 kernel with a stride of one on a single feature map. Unlike the original SPP Redmon and Farhadi (2018), where pooling operations are performed in parallel, SPPF stacks pooling into sequential operations, resulting in faster inference and reduced memory consumption while still aggregating spatial context across multiple receptive fields. After the pooling operations, the resulting feature maps are concatenated along the channel dimension and subsequently passed through a 1×1 convolutional layer. This step reduces dimensionality and promotes effective feature fusion. By encoding object information at various scales, the SPPF module significantly improves detection performance, particularly for objects exhibiting substantial size variability. The SPPF module achieves an optimal balance between accuracy and efficiency, making it an essential component of our crack detection framework. This Backbone configuration ensures consistent extracted feature map dimensions while also hierarchically capturing fine-grained patterns and structural details that are critical for crack detection.

The Neck in YOLOv8 is composed of the combination of a Feature Pyramid Network (FPN) and a Path Aggregation Network (PAN). Broadly, the *FPN* helps the YOLOv8 model to perform multi-scale feature mapping for spatial details and rich semantic context. On the other hand, *PAN* helps to localize the object proposal more accurately, corresponding to object scale and position. The *FPN* (Top-Down) module injects semantic information from deep layers into finer-resolution features for small object detection using the *C2f*, *Concat*, and *Upsample* operations. On the other hand, *PAN* (Bottom-Up) passes spatial detail back down to deeper layers for improved context and large-object detection, utilizing *Downsampling* through strides and *C2f* operations.

The Detection Head in YOLOv8 has both regression and classification branches, as illustrated in Figure 4. It plays a crucial role in generating prediction data, such as object localization and classification, and it does this by leveraging an anchor-free mechanism. Unlike traditional anchor-based methods that rely on predefined anchor boxes, YOLOv8 considers each spatial location (or grid cell) in the output feature map as a potential object center—given a feature map of size $H \times W \times C$, each grid cell outputs a fixed set of predictions: the objectness score (p_o), the class probabilities for each category (p_c), and the bounding box offsets (x, y, w, h). The coordinates (x, y) represent the center of the predicted object relative to the corresponding grid cell, while w and h are the width and height of the expected object. We can then formulate the bounding box coordinates in Equation 1.

$$x = \sigma(\hat{x}) + c_x, \qquad y = \sigma(\hat{y}) + c_y, \qquad w = e^{\hat{w}}, \qquad h = e^{\hat{h}} \tag{1}$$

Where $(\hat{x}, \hat{y}, \hat{w}, \hat{h})$ are the raw outputs of the model, σ is the sigmoid activation that ensures the localization within the cell, and (c_x, c_y) denote the top-left offset of the grid cell in the image space. As shown in Figure 2, we perform predictions at different scales (e.g., P3, P4, P5), enabling effective multi-scale detection. The final predictions undergo Non-Maximum Suppression (NMS) to eliminate redundant bounding boxes, retaining only the most confident detections. After we have the classification score for each category from p_c, we make the final class prediction using the *argmax* function on p_c. This anchor-free approach simplifies training, reduces hyperparameter tuning, making it highly suitable for dense and varied object detection tasks such as pavement distress analysis.

$$\mathcal{FL} = \sum_{p,y} \mathcal{FL}(p, y), \qquad \mathcal{FL}(p, y) = -\sum_{c=1}^{C} \alpha_c \, (1 - p_c)^{\gamma} \, y_c \, \log(p_c), \qquad \alpha_c = \frac{1}{N_c} \tag{2}$$

The Loss Function in the *MoPac+* architecture addresses class imbalance through (i) hard-example mining using Focal Loss to replace standard Cross-Entropy Loss, and (ii) class-weighted loss calculation where weight $\alpha_c = \frac{1}{N_c}$

(inverse of instance count per class). The class weights α_c are normalized to prevent gradient explosion. The focal loss ($\mathcal{FL}$) in Equation 2 is the sum of the weighted cross-entropy loss function over all classes c. The p_c represents the model's estimated probability for class c, and γ is the focusing parameter.

$$\mathcal{L}(y, \hat{y}) = W_c * \mathcal{FL} + W_r * \mathcal{RL}, \qquad \mathcal{RL} = \frac{1}{N} \sum_{i=1}^{N} \left(\|\mathbf{b}_i(x, y, w, h) - \mathbf{b}_i(\hat{x}, \hat{y}, \hat{w}, \hat{h})\|_1 \right) \tag{3}$$

The overall loss function ($\mathcal{L}$) is defined in Equation 3 as the sum of the weighted classification elements with W_c, and regression loss elements weighted with W_r. The regression task for crack detection is less complex than the classification task, thus $W_r << W_c$ in the implementation. Not that the Regression Loss ($\mathcal{RL}$) is the sum of all offsets between the ground truth bounding box $\mathbf{b}_i(x, y, w, h)$ and the predicted box $\mathbf{b}_i(\hat{x}, \hat{y}, \hat{w}, \hat{h})$ for each instance i in Equation 3.

4. MoPac+ AI Ready Datasheet

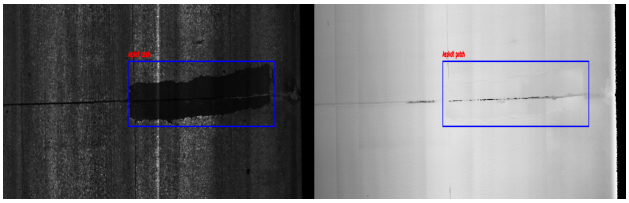


Figure 5: A sample of 2D Intensity, surface-level detail, (left) and 3D Range, depth, (right) pavement surface images with an Asphalt Patch annotated.

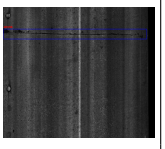
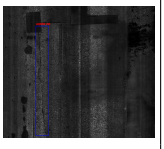
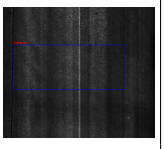
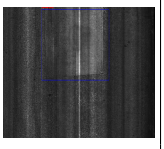
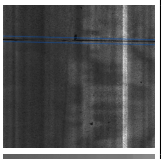
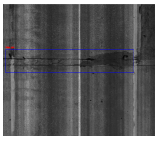
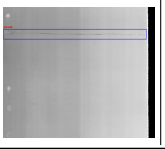
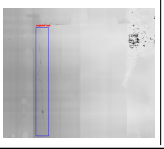
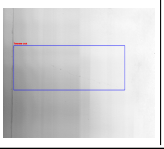
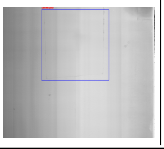
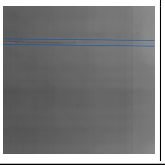
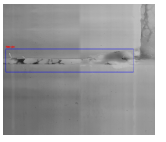
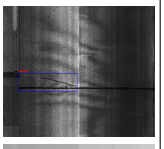
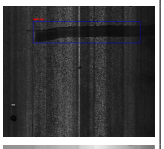
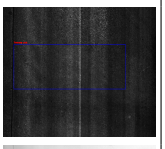
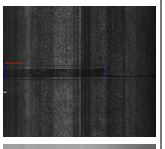
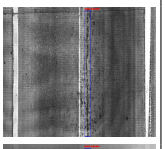
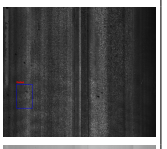
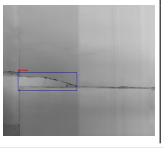
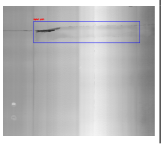
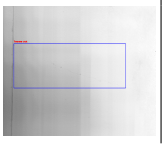
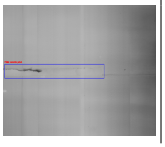
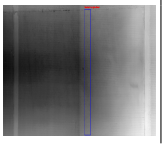
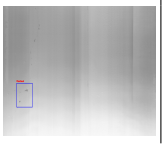
The growing importance of AI in pavement condition assessment necessitates systematic approaches to dataset creation and documentation. Gebru, Morgenstern, Vecchione, Vaughan, Wallach, III and Crawford (2021) pioneered the concept of "datasheets for datasets" to standardize transparency in machine learning data pipelines, emphasizing provenance, collection conditions, and preprocessing steps. In this section, we outline several data requirements to make similar pavement overhead image data AI-ready. The *MoPac+* AI-Ready Datasheet is envisioned as a structured methodology for curating and processing pavement imagery that explicitly addresses the challenges of training robust DNNs for real-world infrastructure monitoring. This approach bridges critical gaps between raw data acquisition and model-ready inputs through three key innovations: (1) sensor-optimized 3D/2D fusion, (2) adaptive pre-processing for distress feature enhancement, and (3) metadata-driven class balancing.

4.1. Data Acquisition and Annotation

The data acquisition system employs a specialized 3D laser camera, capturing simultaneous 2D (Intensity) and 3D (Range) values via a synchronized line-scan camera. This configuration adheres to AASHTO 2D/3D pavement format standards AASHTO (2023), ensuring consistency across segments that cover the full lane width (approximately 4 meters) and a longitudinal range of 2 to 12 meters, depending on driving speed. The triangulation-based depth calculation can produce temporally aligned 2D/3D pairs that preserve spatial relationships critical for crack geometry analysis. After scanning a segment of road, the system then converts the raw data to 2D and 3D grayscale value images— based on various conditions (e.g., vehicle speed)— and stores them in an industry-standard format, where both image types share the same resolution and spatial coordinates within a single file of 1536 x 900 pixels (w x h). The inclusion of 3D height maps enables unique advantages over conventional 2D approaches, as illustrated in Figure 5 with a sample of a 2D and corresponding 3D image of the same road segment.

The critical aspect of making the data AI-ready is the definition of labels that are both semantically meaningful to domain experts and compatible with computer vision algorithms. This dual focus ensures that the labeled data reflects expert knowledge and facilitates efficient training and accurate inference by deep learning models. After data acquisition, the 2D and 3D images were subsequently annotated for common JCP distress classifications by utilizing in-house software developed by Dr. Gong. During this process, the annotators were given access to 2D and 3D imagery and a Google Street View of specific pavement locations. They were then tasked with annotating bounding boxes, delimited by distress classification, around the entire extent of distress classes— including the background— as tightly as possible. During this stage, precise, consistent, and unbiased annotation is paramount to DNN model success. The software then converts these annotations to a YOLO annotation format Jocher, Chaurasia and Qiu (2023) for storage,

Table 1
Displays examples of the Recurring classes (top row) and Rare classes (bottom row) within the JCP dataset, providing both Intensity (2D) and Range (3D) imagery along with their annotations (blue bounding boxes). Notably, the same annotations are used in both the 2D and 3D images.

JCP Recurring Classes: Number of labeled instances is enough for the robust modeling						
Label →	Joint	L-Crack	T-Crack	C-Patch	S-Edge	F-Joint
Instances (Orig.)	4,127	1,097	649	875	8357	519
Instances (After Aug.)	4,527	1,165	668	1562	9063	624
Description	Joint Crack	Longitudinal Crack	Transversal Crack	Concrete Patch	Slab Edge	Failed Joint
Intensity						
Range						
JCP Rare Classes: Number of labeled instances is not enough for robust modeling						
Label →	C-Break	A-Patch	S-Trans	FC-Patch	S-Long	P-Out
Instances (Orig.)	300	346	50	42	441	96
Instances (After Aug.)	300	400	85	72	617	192
Description	Corner Break	Asphalt Patch	Sealed Transverse	Failed Concrete Patch	Sealed Longitudinal	Punchout
Intensity						
Range						

modification, and subsequent AI model input. This study selects 20 highway sections with Joint Concrete Pavement (JCP) surfaces, totaling 7,750 pairs of 2D and 3D images containing 16,943 annotations (or "instances"), titled *JCPv3* (Figure 6). *JCPv3* contains 7,079 image pairs with annotated distress and 671 pairs with no distress present, and all of the images are the same standard dimension of 1536 x 900 pixels (w x h). After initial experiments and evaluation from studies like *MoPac* it was determined that there need to be more robust classes and that some classes too closely match other classes. For these reasons, the major differences between *JCPv1* and *JCPv3* were that: Spall was removed, distributed across other classes; S-Trans was added; Joint was divided into Joint and S-Edge for visual similarity.

4.2. Data Preparation

The *JCPv3* dataset consists of 14 distinct classes of distress, but, due to the limitation of Deep Learning requiring a minimum amount of examples to perform predictably, the distress classes D-Cracking (5 annotations) and Popout (39 annotations) were removed from this implementation. Table 1 illustrates how we split the distress classes into *Recurring* (over 500 instances) and *Rare* (under 500 instances) categories, with representative 2D intensity images and corresponding 3D range images provided for each class. This class splitting reveals the critical challenges inherent to pavement distress detection: (i) Recurring classes (e.g., Slab Edge) dominate the distribution, while rare classes (e.g.,

Algorithm 1: Stratified dataset splitter with per-class balancing

```
Don't Print Semicolon   Function stratified_split(class_to_files, min_per_class, test_ratio):
    Initialize empty set: test_files;
    foreach class, files in class_to_files do
        Shuffle file list;
        num_test ← min(length, max(min_per_class, test_ratio * total));
        Add first num_test files to test_files;
    return test_files;
Function split_dataset(image_dir, annotation_dir, output_root, test_ratio):
    Set random seed to ensure reproducibility;
    Create 'train/images', 'train/labels', 'test/images', and 'test/labels' directories under output_root;
    class_to_files ← load_class_instances(annotation_dir);
    test_files ← stratified_split(class_to_files, min_per_class=20, test_ratio);
    foreach img_file in image_dir do
        if file does not end with '.png' then
            continue
        base ← filename without extension;
        ann_file ← base + '.txt';
        if img_file is in test_files then
            split ← 'test'
        else
            split ← 'train'
        Copy img_file to split/images;
        Copy ann_file to split/labels;
```

Punchout) exhibit severe under-representation ($\leq$ 8% of total instances); (ii) Longitudinal cracks and joint features demonstrate $\leq$ 12% visual similarity in gradient histograms, confounding traditional detection approaches.

Dataset Stratification The dataset was partitioned into training and testing subsets, ensuring that at least 20 instances represented each class in the test set. This threshold was determined based on the statistical distribution of the dataset, thereby supporting reliable and meaningful model evaluation. Algorithm 1 presents the pseudo-code for our stratified train-test split procedure. Initially, we enumerate the instances per class using the *load_class_instances* method. We then apply a stratification strategy based on two criteria: (1) a minimum number of instances per class (`min_per_class`), and (2) a specified test set ratio (`test_ratio`), selecting the greater of the two for each class. This approach ensures a balanced representation of both common and rare classes in the test set, facilitating standardized benchmark comparisons and enhancing the dataset's suitability for AI-driven analysis.

Image Augmentation The augmentation increases the sample size and makes the crack lines thicker and more visible (Note: We perform augmentations only for rare classes with fewer than 200 instances). We perform a series of augmentations on the images. However, we rejected any types of morphological augmentation that might significantly alter the shape or size of the cracks. The augmentations used in the pre-processing are as follows: 1) Horizontal Flip, 2) Gaussian Noise, 3) CLAHE, 4) CANNY, 5) Random Brightness Contrast, and 6) Elastic Transform. We selected four random augmentations from these six types for pre-processing. This augmentation process improved the overall model performance for the test dataset; the results are presented later in the ablation study in section 5.

5. Experiments

In this section, we present a comprehensive evaluation of our proposed *MoPac+* model against state-of-the-art approaches using our experimental dataset. Our dataset comprises 12 distinct classes, meticulously pre-processed through the augmentation pipeline detailed in Section 4. To ensure rigorous evaluation, we employ an instance-aware

Table 2
Table for the average performance of different SOTA Models for all 12 classes using 2D images only.

Anchor	Algorithm	Prec.	Recall	mAP 50	mAP 50-95	L. Param (M.)	GFLOPS
Yes	YOLOv5m	0.68	0.54	0.60	0.32	**21.22**	**49.50**
Yes	YOLOv7	0.67	0.54	0.55	0.28	36.97	104.70
No	CenterNet2	0.69	0.58	0.64	0.32	58.10	210.20
No	DEFORM. DETR	0.68	0.55	0.62	0.31	42.48	173.63
No	YOLOv8m (Baseline)	0.67	0.59	0.62	0.34	25.85	78.70
No	*MoPac+*	**0.71**	**0.67**	**0.69**	**0.38**	25.85	78.70

train-test splitting strategy as formalized in Algorithm 1. Additionally, we maintained consistent values for common hyperparameters, such as learning rate, epochs, and IOU, for all comparison methods.

5.1. Setup

The *MoPac+* architecture is implemented in Python, extending the official YOLOv8 implementation from ULTRALYTICS within the PyTorch deep learning framework. All experiments are conducted on hardware featuring an NVIDIA Titan XP 1080 GPU (12GB VRAM), an Intel Core i9-11900K processor (3.50 GHz, 16 cores), and 167GB DDR4 system memory. We maintain consistent training parameters across all experiments with 25 training epochs, input size=640, learning rate=0.0001, and batch size=16. For *MoPac+*, the loss weights are set to $W_r = 0.1$ (regression) and $W_c = 0.5$ (classification) and class and box loss gain=0.5. Our analysis employs five principal metrics to assess detection performance:

- **Recall (R)** - Ratio of correctly identified positive instances to all actual positives.
- **Average Recall (AR)** - Recall averaged across multiple Intersection-over-Union (IoU) thresholds.
- **Precision (P)**: Proportion of correct detections among all positive predictions.
- **Average Precision (AP)** - Area under the Precision-Recall curve per class, as defined in Equation 4.
- **Mean Average Precision (mAP)** - Mean AP across all classes, computed at a certain IoU threshold, as defined in Equation 4.

$$\text{mAP} = \frac{1}{C}\sum_{c=1}^{C} \text{AP}_c, \qquad \text{AP}_c = \int_0^1 p_c(r)\,dr \quad \text{(Precision-Recall area for class } c\text{)} \tag{4}$$

The mAP metric serves as our primary benchmark for model comparison, with class-specific AP scores providing detailed insights into detection capabilities across both common and rare categories. To address class imbalance, we stratify our evaluation into two distinct groups: (i) *Frequent Classes*: Commonly occurring categories with abundant training samples, and (ii) *Rare Classes*: Infrequently observed categories requiring specialized analytical consideration. This stratification enables separate analysis of model performance on high-frequency classes ($N_c \geq \tau$) and low-frequency classes ($N_c < \tau$), where N_c represents the instance count for class c and τ is our frequency.

5.2. Comparison with State-of-the-Art Methods for Overall Performance

Table 2 presents a comparative evaluation of state-of-the-art (SOTA) object detection models, alongside the proposed *MoPac+* framework. It is important to note that YOLOv5 and YOLOv7 Redmon et al. (2016); Redmon and Farhadi (2017) utilize anchor-based detection mechanisms, whereas CenterNet2, DETR, YOLOv8m (serving as the baseline), and *MoPac+* operate without anchors. This study incorporates both anchor-based and anchor-free approaches to rigorously assess the efficacy of anchor-free detectors in accurately identifying objects with diverse shape characteristics. From the results summarized in Table 2, YOLOv5 achieves a competitive Precision score of 0.68 relative to YOLOv7 and YOLOv8. The CenterNet2 becomes the closest competitor with an overall AP of 0.69, and an mAP(IOU=0.5) of 0.64. The transformer-based DETR performs very close to CenterNet2 in terms of Prec., however, the recall drops by $\Delta 3\%$. The proposed *MoPac+* model attains the highest Precision, surpassing the nearest SOTA

(CenterNet2) by 2%. Regarding Recall and mean Average Precision (mAP) at IoU=0.50, the CenterNet2 model is the closest competitor. Specifically, *MoPac+* achieves a Recall of 0.67 and mAP@0.50 of 0.69, representing improvements of 9% in Recall and 5% in mAP compared to CenterNet2. Notably, anchor-free methods yield higher Recall values and significantly outperform anchor-based detectors in terms of mAP. Additionally, the observed mAP@0.50 exceeds the corresponding mAP averaged over IoU thresholds 0.50 to 0.95, consistent with prior findings reported in Warren, Garrard, E. and Tešić, J. (2018) for overhead imagery. Figure 7 illustrates the qualitative result comparison for MoPac+ with relevant SOTA models. Figure 7 (fourth row) illustrates two instances of punchout distress, which are visually distinct from a computer vision perspective. This diversity poses a classification challenge for most methods. Importantly, the proposed *MoPac+* model successfully detects both instances correctly, attributable to its robust feature extraction backbone, extensive data augmentation during training, and the integration of hard-example mining techniques. Our proposed model maintains a balanced performance gain regardless of the distress sizes and the recurrence in the dataset.

As one of our primary motivations was to propose a model that is lightweight enough to deploy in resource-constrained devices, in Table 2 we present the L. Params and GFLOPS for comparing SOTA models. From Table 2 we see that the YoloV5 requires the lowest computational resource. On the other hand, despite competitive performance, the CenterNet2 requires the highest amount of computational resources. We decided to use YOLOv8m as the baseline model because it shows satisfactory performance for road crack detection and requires moderately less computational power. The inference rate for CenterNet2 is 45.8ms/frame; on the other hand, YOLOv8 is 1.8ms/frame, which is a standard real-time response for common computer vision applications.

5.3. Class-wise Performance Comparison

In this section, we conduct the class-wise analysis based on the *Recurring* and *Rare* distress category classifications as defined previously in Section 4 and Table 1. Using Average Precision (AP) and Average Recall (AR), Table 3 shows the performance of *Recurring* and *Rare* classes with SOTA and proposed *MoPac+* methods.

Recurring Classes: The dominant performance comes from our proposed *MoPac+* model (See Table 3), with YOLOv7 being the closest competitor in detecting the recurrent courses. YOLOv5 demonstrates superior performance in T-Crack detection, achieving the highest AP of 0.74 and exhibiting the best precision for F-Joint detection among SOTA models at 0.61. However, its AR performance varies considerably across crack types. On the other hand, the lightweight DEFORM. DETR achieves the competitive AP and the highest AR (0.85) for the L-Crack class. YOLOv7 performs well in terms of Recall, achieving the highest AR scores for C-Patch (0.86) and S-Edge (0.95) detection. YOLOv7's strength lies in identifying common instances of the distress types at the expense of lower precision. *MoPac+* achieves the highest AP scores across five out of six crack types: J-Crack (0.94), L-Crack (0.73), C-patch (0.83), S-Edge (0.97), and F-Joint (0.62). For distress types such as S-Edge and J-Crack, we achieved nearly perfect precision results using our *MoPac+* model. Thus, even in a detailed evaluation of recurrent distress types presented in Table 3, our method outperforms the available SOTA methods for our dataset.

Rare Classes: The findings from Table 3 were interesting and closely aligned with our hypothesis regarding the use of weighted loss and hard-example mining for these classes. For Rare classes, the shapes can be challenging, and the annotation instance count was very limited. YOLOv5 demonstrates strong Precision capabilities, achieving the highest AP for C-Break detection (0.60) and competitive performance on A-Patch (0.64), S-Trans (0.95), and FC-Patch (0.90). YOLOv7 shows exceptional Precision performance for S-Trans (0.98) and FC-Patch (0.98), achieving the highest AP across all methods. Despite this, YOLOv7 did not perform well for these classes as reflected in its AR scores in S-Trans (0.18) and FC-Patch (0.18), indicating its limitations when training with a small sample size. The CenterNet2 Zhou, Koltun and Krähenbühl (2021) is a point-based network efficient for pixel-level feature extraction. Hence, it shows relatively better performance for Small-sized classes, such as FC-Patch and P-Out, achieving an AR of 0.85 and an AP of 0.64, respectively—the DEFORM. DETR also shows promising performance for small classes and remains the closest competitor compared to the MoPac+ method. Our *MoPac+* method outperformed competitors for five out of six crack types and achieved the highest Recall across multiple categories: C-Break (0.49), A-Patch (0.66), S-Trans (0.70), FC-Patch (0.30), and S-Long (0.90).

5.4. Ablation Study

This section focuses on an ablation study with the purpose of determining the various effects the different enhancements of our proposed method have on DNN performance. Using our baseline of YOLOv8, we incrementally integrate the improvements of the *MoPac+* pipeline and explain the rationale behind the method by analyzing its

Table 3
Performance of the SOTA models per recurring label (top) and rare label (bottom) on the test set. Here, AP and AR were used as the evaluation metrics.

Recurrent Label	J-Crack		L-Crack		T-Crack		C-Patch		S-Edge		F-Joint	
Method ↓	AP	AR	AP	AR	AP	AR	AP	AR	AP	AR	AP	AR
YOLOv5m	0.85	0.86	0.64	0.65	**0.74**	0.55	0.72	0.83	0.84	0.90	0.61	0.67
CenterNet2	0.71	0.42	0.69	**0.85**	0.66	0.63	0.77	0.80	0.80	0.91	0.61	0.50
DEFORM. DETR	0.73	0.80	0.66	0.78	0.67	**0.68**	0.70	0.81	0.84	0.89	0.58	0.47
YOLOv7	0.69	0.18	0.62	0.76	0.70	0.64	0.67	**0.86**	0.83	**0.95**	0.60	0.56
YOLOv8s	0.79	0.87	0.60	0.68	0.67	0.57	0.70	0.81	0.89	0.93	0.49	0.73
YOLOv8m (Baseline)	0.80	**0.87**	0.60	0.73	0.67	0.60	0.74	0.84	0.91	0.93	0.49	0.75
MoPac+	**0.94**	0.73	**0.73**	0.67	0.64	0.58	**0.83**	0.77	**0.97**	0.80	**0.62**	**0.75**
Rare Label	C-Break		A-Patch		S-Trans		FC-Patch		S-Long		P-Out	
Method ↓	AP	AR	AP	AR	AP	AR	AP	AR	AP	AR	AP	AR
YOLOv5m	0.60	0.32	0.64	0.54	0.95	0.20	0.90	0.18	0.73	0.82	0.50	0.47
CenterNet2	0.55	0.40	0.70	0.56	0.88	0.30	0.97	0.26	0.74	0.85	**0.64**	0.63
DEFORM. DETR	**0.62**	0.42	0.60	0.55	0.75	0.25	0.90	0.30	0.71	0.86	0.58	0.59
YOLOv7	0.58	0.41	0.47	0.57	**0.98**	0.18	**0.98**	0.18	0.62	0.87	0.24	**0.72**
YOLOv8s	0.48	0.33	0.72	0.56	0.12	0.22	0.44	0.18	0.74	0.80	0.35	0.28
YOLOv8m (Baseline)	0.50	0.33	**0.74**	0.56	0.12	0.25	0.44	0.16	0.76	0.80	0.38	0.36
MoPac+	0.50	**0.49**	0.62	**0.66**	0.45	**0.70**	0.90	**0.30**	**0.83**	**0.90**	0.36	0.40

performance and impact on the overarching modeling process. We chose YOLOv8 with a CSPDarknet53 backbone because of its superior performance for a range of shapes and better performance for small objects. The baseline model achieves a precision of 0.668 and a recall of 0.592, respectively. The mAP from the baseline is 0.621, which gives a good starting point. Next, as we observe from Table 1, we have several classes with few samples. To address this issue, we performed a series of pre-processing steps on rare courses. It is evident from Table 4 that pre-processing helps improve recall and mAP by at least +1.2%. The shapes of distress can be clustered into very pre-processing steps, so the regression task is less complex in our applications. We aim to identify the types of distress that are challenging, as discussed in earlier sections. To resolve this issue, we perform weight balancing on the regression and classification pipeline. We achieved a significant improvement after performing weight balancing, with nearly a 2% increase in all evaluation matrices. In Figure 8, we plot the effectiveness of the Weight balance technique with YOLOv8. From this figure, we see that the weight balance is forced to classify and detect more classes. To address the challenges of rare class detection, we implemented a class-balanced focal loss function that dynamically penalizes misclassifications based on the frequency of class instances in the training dataset. Our hard-example mining strategy yielded significant improvements, achieving +1.5% mAP and a +2.2% recall gain for rare classes compared to baseline approaches. Figure 8 provides qualitative visualizations of this improvement through comprehensive ablation studies. We also investigated alternative classification paradigms, including support vector machines (SVM), but found Cross-Entropy with hard-example mining demonstrated superior performance for this specific task. To optimize post-processing, we empirically evaluated Non-Maximum Suppression (NMS) thresholds across 0.3, 0.5, 0.7 IOUs. Our analysis revealed very similar performance between 0.3 and 0.5 thresholds ($\Delta mAP < 0.1\%$), with an IOU of 0.5 emerging as optimal for distress detection. Using this configuration, our *MoPac+* framework achieved state-of-the-art performance with 0.692 mAP and 0.665 recall, establishing a new SOTA benchmark for imbalanced pavement distress detection tasks.

6. Conclusion and Future Work

In this paper, we propose *MoPac+*, a state-of-the-art crack localization and identification framework that effectively recognizes different types of cracks in the pavement from intensity images. We have prepared an AI-ready dataset to perform a detection benchmark task across six recurrent and six rare distress types. First, we have significantly improved the labeling of our training dataset, region annotation, and defined categories that are visually meaningful for both human experts and the automated system. We introduce the anchor-free YOLOv8 pipeline that robustly detects joints, longitudinal and traversal cracks, and asphalt patches captured over larger patches of the road. We have employed the weight-balance technique to place greater emphasis on the classification task than on the regression task. Next,

Table 4
Ablation study for different modules of our *MoPac+* method. With the columns representing: Baseline (YOLOv8m), Data Augmentation (D Aug.), Focal Loss (F. Loss)— used for hard-example mining—, Weight Balance (Weight B)— the balance between classification and regression loss—, and Support Vector Machine (SVM).

Baseline	Data Aug.	F. Loss	Weight B	SVM	Prec.	Recall	mAP 50	mAP 50-95
w CSPDNet53					0.668	0.592	0.621	0.335
w Aug	✓				0.657	0.601	0.635	0.345
w Weight B.	✓	✓			0.673	0.626	0.667	0.363
w F.Loss	✓	✓	✓		0.685	0.644	0.670	0.374
w SVM	✓	✗	✓	✓	0.653	0.604	0.658	0.353
MoPac+ $IOU = 0.3$	✓	✓	✓	✗	0.701	0.665	0.690	0.381
MoPac+ $IOU = 0.5$	✓	✓	✓	✗	**0.705**	**0.665**	**0.692**	**0.381**
MoPac+ $IOU = 0.7$	✓	✓	✓	✗	0.685	0.644	0.670	0.374

we propose hard-example mining with focal loss to improve performance on rare classes. Using IOU=0.5, we have achieved the highest recall of **0.665** and the highest mean average precision of **0.692**, respectively, outperforming the latest state-of-the-art (SOTA) methods. The MoPac+'s next step is to continue developing high-quality training datasets and improve detection performance using multimodal data, such as range images and thermal images, for context-aware feature extraction.

AUTHOR CONTRIBUTIONS

Debojyoti led the machine learning implementation, data augmentation, and experiments, and contributed to the preparation of the manuscript. Scouten created tables and figures and contributed to manuscript preparation and editing. Gong led the creation and manual annotation of the dataset, created figures, and contributed to the literature review, manuscript preparation, and editing. Tešić designed the experiments and data analysis, and contributed to the literature review, the manuscript preparation, and editing. Wang conceived the research idea, supervised the project, and contributed to the manuscript writing and editing.

Acknowledgments

The study has been partially supported by funding from the following sources: TxDOT Project #0-7150 and NSF Project 2213694.

References

AASHTO, 2023. Standard specification for file format of two-dimensional and three-dimensional (2d/3d) pavement image data.

Biswas, D., Tešić, J., 2022. Small object difficulty (sod) modeling for objects detection in satellite images, in: 2022 14th International Conference on Computational Intelligence and Communication Networks (CICN), IEEE. pp. 125–130.

Biswas, D., Tešić, J., 2024. Binarydnet53: a lightweight binarized cnn for monkeypox virus image classification. Signal, Image and Video Processing 18, 7107–7118.

Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M., 2020. Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 .

Cai, Z., Vasconcelos, N., 2019. Cascade r-cnn: High quality object detection and instance segmentation. IEEE transactions on pattern analysis and machine intelligence 43, 1483–1498.

Cao, J., Yang, G., Yang, X., 2020. Pavement crack detection with deep learning based on attention mechanism. Journal of Computer-Aided Design & Computer Graphics 32, 1324–1333.

Chen, X., Liu, C., Chen, L., Zhu, X., Zhang, Y., Wang, C., 2024. A pavement crack detection and evaluation framework for a uav inspection system based on deep learning. Applied Sciences 14, 1157.

Deng, L., Yuan, H., Long, L., Chun, P.j., Chen, W., Chu, H., 2024. Cascade refinement extraction network with active boundary loss for segmentation of concrete cracks from high-resolution images. Automation in Construction 162, 105410.

Gavilán, M., Balcones, D., Marcos, O., Llorca, D.F., Sotelo, M.A., Parra, I., Ocaña, M., Aliseda, P., Yarza, P., Amírola, A., 2011. Adaptive road crack detection system by pavement classification. Sensors 11, 9628–9657.

Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J.W., Wallach, H., III, H.D., Crawford, K., 2021. Datasheets for datasets. Commun. ACM 64, 86–92. URL: https://doi.org/10.1145/3458723, doi:10.1145/3458723.

Gong, H., Tešić, J., Tao, J., Luo, X.H., Wang, F., 2023. Automated pavement crack detection with deep learning methods: What are the main factors and how to improve the performance? Transportation Research Record 2677, 311–323. doi:10.1177/03611981231161358.

He, K., Gkioxari, G., Dollár, P., Girshick, R., 2017. Mask r-cnn, in: Proceedings of the IEEE international conference on computer vision, pp. 2961–2969.

Hu, G., Hu, B., Yang, Z., Huang, L., Li, P., 2021. Pavement crack detection method based on deep learning models. Wireless Communications and Mobile Computing 2021. doi:10.1155/2021/5573590.

Hu, N., Yang, J., Jin, X., Fan, X., 2023. Few-shot crack detection based on image processing and improved yolov5. Journal of Civil Structural Health Monitoring 13, 165–180.

Huyan, J., Li, W., Tighe, S., Zhai, J., Xu, Z., Chen, Y., 2019. Detection of sealed and unsealed cracks with complex backgrounds using deep convolutional neural network. Automation in Construction 107, 102946.

Jocher, G., Chaurasia, A., Qiu, J., 2023. Ultralytics YOLO. URL: https://github.com/ultralytics/ultralytics.

Jocher, G., the Ultralytics team, 2023. YOLOv8: Ultralytics Official Implementation. https://github.com/ultralytics/ultralytics. Accessed: 2025-06-07.

Kang, D.H., Cha, Y.J., 2022. Efficient attention-based deep encoder and decoder for automatic crack segmentation. Structural Health Monitoring 21, 2190–2205.

Li, D., Xie, Q., Gong, X., Yu, Z., Xu, J., Sun, Y., Wang, J., 2021. Automatic defect detection of metro tunnel surfaces using a vision-based inspection system. Advanced Engineering Informatics 47, 101206.

Li, Q., Liu, X., 2008. Novel approach to pavement image segmentation based on neighboring difference histogram method, in: 2008 congress on image and signal processing, IEEE. pp. 792–796.

Liu, F., Xu, G., Yang, Y., Niu, X., Pan, Y., 2008. Novel approach to pavement cracking automatic detection based on segment extending, in: 2008 International Symposium on Knowledge Acquisition and Modeling, IEEE. pp. 610–614.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C., 2016. Ssd: Single shot multibox detector, in: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14, Springer. pp. 21–37.

Pham, V., Nguyen, D., Donan, C., 2022. Road damage detection and classification with yolov7, in: 2022 IEEE International Conference on Big Data (Big Data), IEEE. pp. 6416–6423.

Redmon, J., Divvala, S., Girshick, R., 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.

Redmon, J., Farhadi, A., 2017. Yolo9000: better, faster, stronger, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7263–7271.

Redmon, J., Farhadi, A., 2018. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 .

Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems 28.

Roy, A.M., Bhaduri, J., 2023. Densesph-yolov5: An automated damage detection model based on densenet and swin-transformer prediction head-enabled yolov5 with attention mechanism. Advanced Engineering Informatics 56, 102007.

Saberironaghi, A., Ren, J., 2024. Depthcracknet: A deep learning model for automatic pavement crack detection. Journal of Imaging 10, 100.

Scouten, A., Gong, H., Tešić, J., Wang, F., 2025. Deep learning pipeline for modeling pavement cracks with an imbalanced data set (mopac). Poster presentation at the Transportation Research Board (TRB) 104th Annual Meeting. URL: https://annualmeeting.mytrb.org/OnlineProgramArchive/Details/22848. presentation Number: TRBAM-25-06415.

Son Dong Nguyen, S.D., Tran, T.S., Tran, V.P., Lee, H.J., Piran, M.J., Le, V.P., 2022. Deep learning-based crack detection: A survey. International Journal of Pavement Research and Technology 16, 943–967. doi:10.1007/s42947-022-00172-z.

Tran, V.P., Tran, T.S., Lee, H.J., Kim, K.D., Baek, J., Nguyen, T.T., 2021. One stage detector (retinanet)-based crack detection for asphalt pavements considering pavement distresses and surface objects. Journal of Civil Structural Health Monitoring 11, 205–222.

Texas Department of Transportation, T., 2023. 2023 roadway inventory report summary mileage, traffic and environmental statistics. URL: https://www.txdot.gov/content/dam/docs/tpp/roadway-inventory/road-inventory-annual-report-2023.pdf.

Warren, N., Garrard, B., E., S., Tešić, J., 2018. Transfer learning of deep neural networks for visual collaborative maritime asset identification, in: 2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC), pp. 246–255.

Wu, P., Wu, J., Xie, L., 2024. Pavement distress detection based on improved feature fusion network. Measurement 236, 115119.

Yaseen, M., 2024. What is YOLOv8: An in-depth exploration of the internal features of the next-generation object detector. arXiv cs.CV URL: https://arxiv.org/abs/2408.15857, arXiv:2408.15857.

Ye, G., Qu, J., Tao, J., Dai, W., Mao, Y., Jin, Q., 2023. Autonomous surface crack identification of concrete structures based on the yolov7 algorithm. Journal of Building Engineering 73, 106688.

Yu, G., Zhou, X., 2023. An improved yolov5 crack detection method combined with a bottleneck transformer. Mathematics 11, 2377.

Zheng, L., Xiao, J., Wang, Y., Wu, W., Chen, Z., Yuan, D., Jiang, W., 2024. Deep learning-based intelligent detection of pavement distress. Automation in Construction 168, 105772.

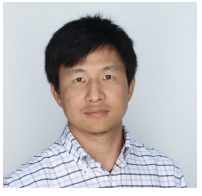
Zhou, S., Yang, D., Zhang, Z., Zhang, J., Qu, F., Punetha, P., Li, W., Li, N., 2025. Enhancing autonomous pavement crack detection: Optimizing yolov5s algorithm with advanced deep learning techniques. Measurement 240, 115603.

Zhou, X., Koltun, V., Krähenbühl, P., 2021. Probabilistic two-stage detection. arXiv preprint arXiv:2103.07461 .

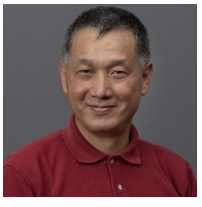
Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J., 2020. Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159 .

Debojyoti Biswas is a Computer Science Ph.D. candidate at Texas State University, US. He received his B.Sc. in Information and Communication Engineering from Noakhali Science and Technology University, Bangladesh. He worked as a Lecturer in the Computer Science department at Leading University, Bangladesh, from 2019 to 2021, before commencing his Ph.D. studies. He is a peer reviewer for several prestigious journals, including IEEE TGRS, IEEE JSTARS, IEEE GRSL, Springer PFGE, Elsevier ESA, and others. His research interests include computer vision, image processing, video understanding, object detection, and domain adaptation.

Andrew Scouten is a graduate student pursuing an M.S. Degree in Computer Science at Texas State University. He began his research in Computer Vision in early 2023 during his undergraduate study at Texas State, where he contributed to the development of the AI systems to detect bacterial adhesion and corrosion in SEM images from NASA's International Space Station. Currently, he is working as a research assistant on projects funded by the Texas Department of Transportation, focusing on the application of artificial intelligence for pavement distress detection and measurement. Andrew's research interests include the interdisciplinary applications of AI and deep learning, as well as explaining and improving existing AI systems.



Haitao Gong, Ph.D. is a postdoctoral scholar at Texas State University, where he also earned his Ph.D. in Materials Science, Engineering, and Commercialization. He began his academic journey with a bachelor's degree in Transportation from Huazhong University of Science and Technology, followed by a master's degree in Transportation Planning and Management from Tongji University in Shanghai. Before his doctoral studies, he worked as a transportation engineer at China Communications Construction Company, focusing on transportation management and planning. His current research interests include pavement management, deep learning, and image processing, with a particular emphasis on AI-driven approaches to infrastructure evaluation.



Feng Wang, PE, Ph.D. is a Professor in Transportation Engineering and Program Coordinator of the Civil Engineering Program of the Ingram School of Engineering at Texas State University in San Marcos, Texas. He also serves as Associate Director of the US DOT Tier I University Transportation Center, CREATE. Feng received his Ph.D. in Transportation Engineering from the University of Texas at Austin. His current research activities include the development of AI/ML-based image processing methods for automated pavement condition evaluation using 2D/3D/ pavement surface images, image-based pavement condition data analysis, and decision support modeling for pavement management systems and pavement warranty programs. He has received research funding from the US DOT, FHWA, NSF, Texas DOT, and Mississippi DOT.

Jelena Tešić, Ph.D. is an Associate Professor at Texas State University. Before joining academia, she was a research scientist at Mayachitra (CA) and the IBM Watson Research Center (NY). She received her Ph.D. (2004) and M.Sc. (1999) in Electrical and Computer Engineering from the University of California, Santa Barbara, CA, USA, and Dipl. Ing. (1998) in Electrical Engineering from the University of Belgrade, Serbia. Her research focuses on advancing AI for scientific applications, including object localization and identification at scale in overhead imagery. NSF, NIH, NAVAIR, DARPA, ONR, USDA, and DoE funded Dr. Tešić work. Dr. Tešić is in the organizing committee of the IEEE ICMR 2027, and she has served as the Area Chair for the IEEE ICIP, IEEE ICME, ACM IMX, and ACM MM area chair, and as the Guest Editor for IEEE Multimedia Magazine. She has authored over 80 peer-reviewed scientific papers and holds seven US patents. She is an IEEE and NAI Senior member.

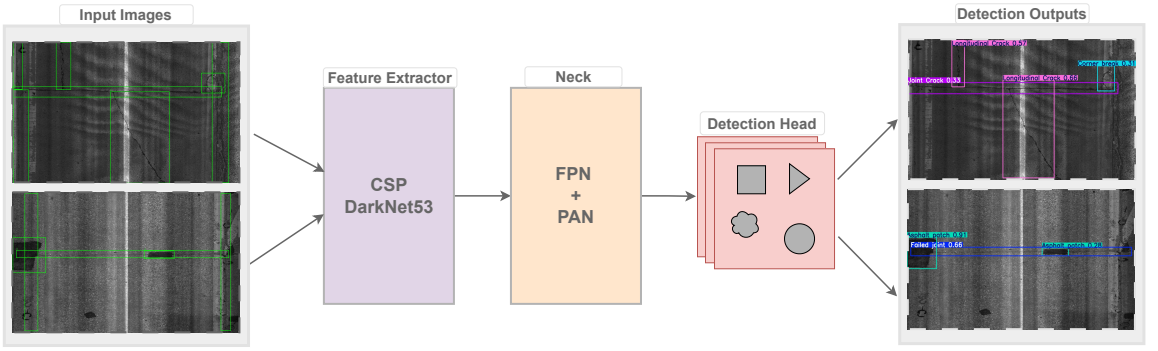


Figure 1: MoPac+ Architecture with three modules: (1) Feature Extractor; (2) the Neck, and (3) Detection Head.

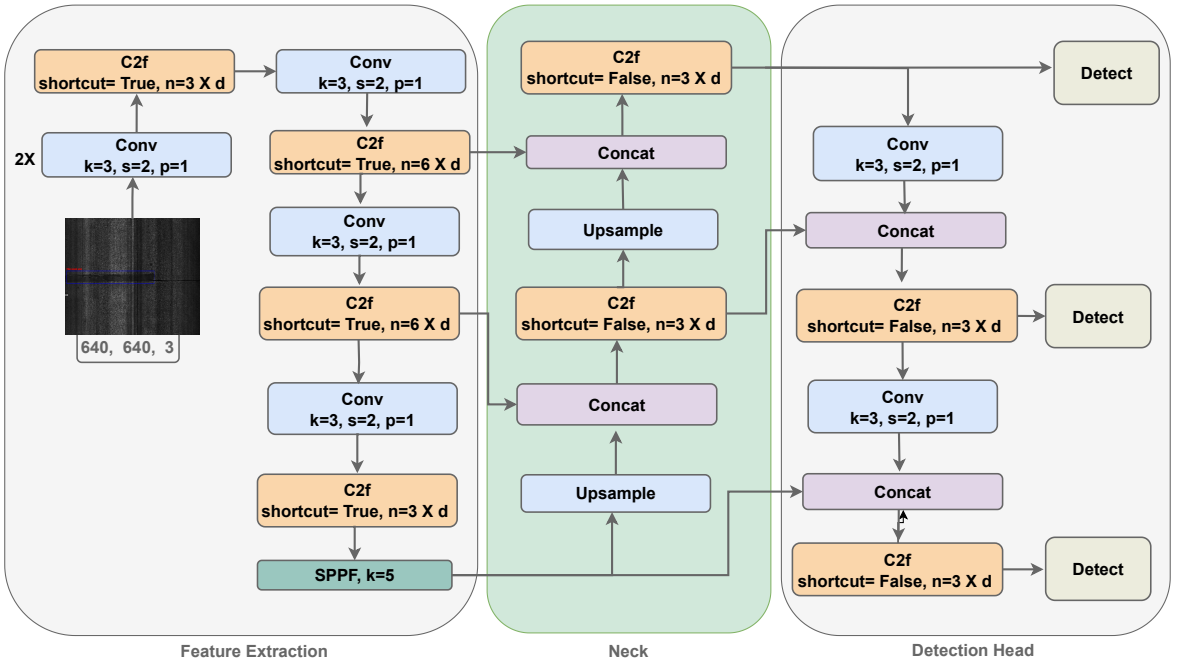


Figure 2: MoPac+: The proposed architecture, abstractly similar to YOLOv8, with detailed implementations of the Feature Extraction (Backbone), Neck, and the Detection Head. The Detection Head has three Detect modules which represent multi-scale prediction on different scales, e.g., P3 (Top), P4 (Middle), and P5 (Bottom).

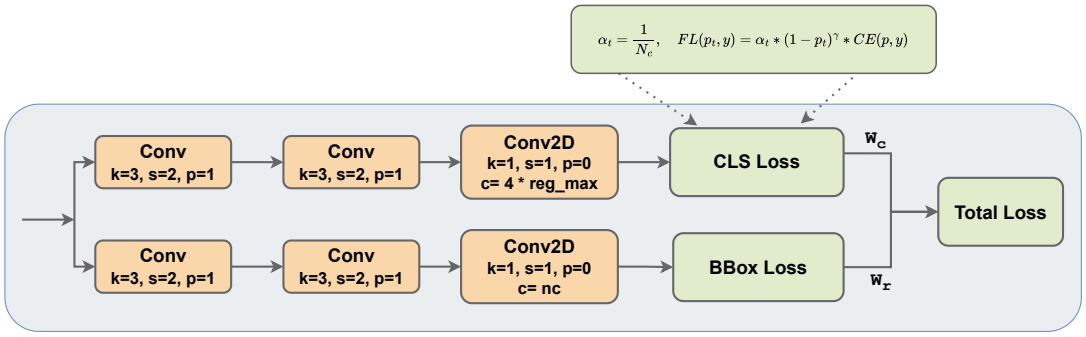


Figure 4: The *Detection Head* has two branches. 1) Classification branch, 2) Regression branch. We perform weight balancing and hard-example mining in this module.

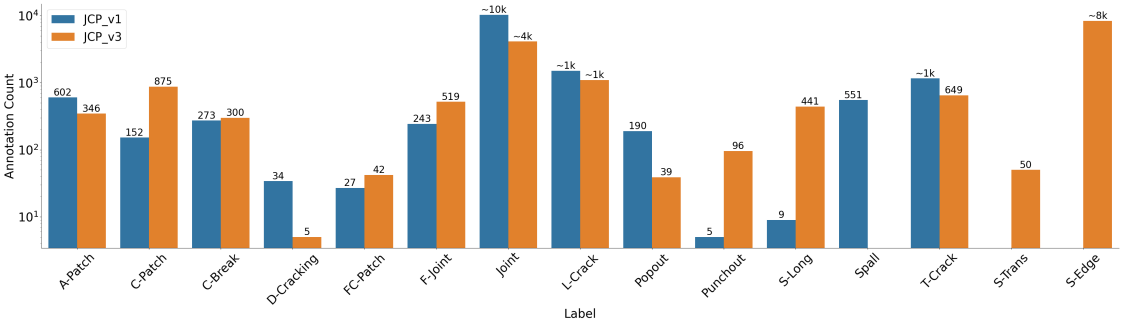


Figure 6: Annotation class counts by JCP dataset version, before any data augmentation, where MoPac uses JCPv1, JCPv2 is an intermediary version, and MoPac+ uses JCPv3.

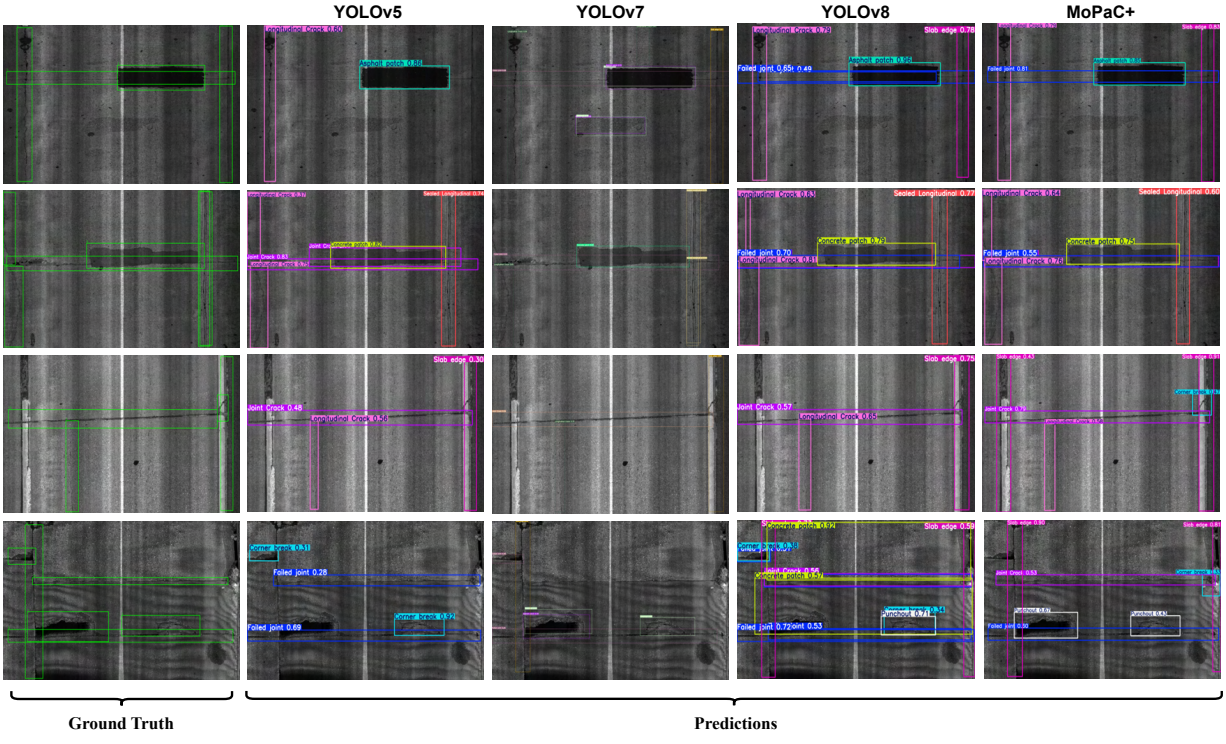


Figure 7: SOTA detection comparison with proposed *MoPac+* model. Depicting model prediction done on four randomly selected test samples (rows), comparing the ground truth, YOLOv5, YOLOv7, YOLOv8, and MoPac+, respectively (columns).

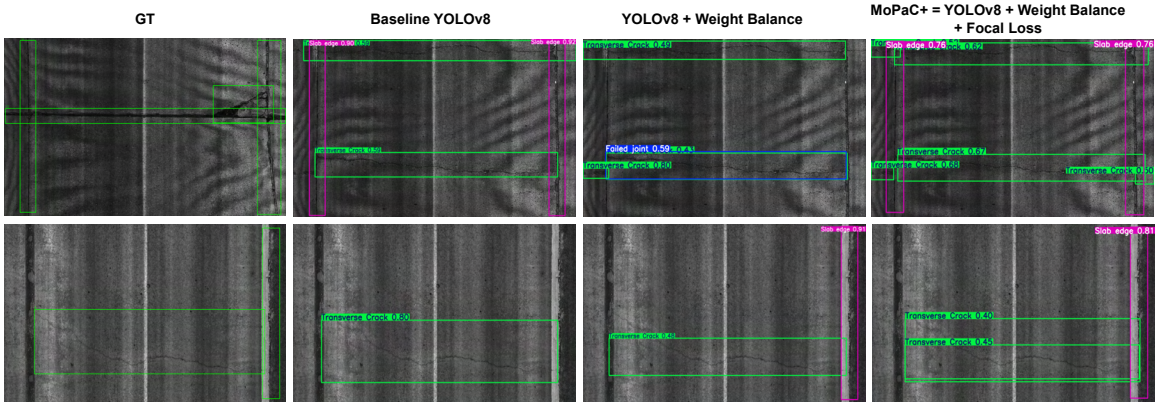


Figure 8: Ablation Study for different modules in the proposed *MoPac+* pipeline. We compare the detection results with those of the baseline YOLOv8 architecture.