Pavement Distress Classification Using UAV Imagery

Technical Project Report

Author: Parth Chaudhari

Date: 2025

1. Overview

This report documents a complete machine learning pipeline for classifying pavement distress in high-resolution UAV runway imagery. The goal is to automatically determine whether a 160×160 pixel image patch contains visible pavement distress such as cracks or surface degradation. The system uses raw UAV images, Pascal VOC XML annotations, automated preprocessing, patch generation, and a TensorFlow-based CNN model to perform binary classification at the patch level.

2. Data Sources

The project uses two primary inputs: (1) JPEGImages, containing 2,426 UAV pavement images, and (2) Annotations, containing Pascal VOC XML files for each image. The XML annotations specify bounding boxes for visible distress, enabling supervised learning.

3. Annotation Parsing

Each XML file was parsed using Python's xml.etree.ElementTree to extract bounding box coordinates for every distress region. This created a mapping from image file paths to lists of distress bounding boxes, forming the basis for constructing the labeled dataset.

4. Patch Generation

The raw images were processed to extract 160×160 patches used as training samples. Positive patches were centered around distress bounding boxes to guarantee visible cracking. Negative patches were randomly sampled from areas with no bounding box overlap, ensuring clean pavement representation. This tile-based approach supports scalable inspection workflows typically used in remote sensing and infrastructure analysis.

Patch Dataset Statistics

A total of 14,370 patches were created. Of these, 11,946 were positive (distressed) patches and 2,424 were negative (healthy) patches. This reflects the natural imbalance present in pavement imagery where cracks occupy relatively small portions of the surface.

5. Dataset Construction

Each patch was indexed in a structured dataset containing its file path, source image, and binary label. A stratified split was applied to produce training (70%), validation (15%), and testing (15%) sets. This preserved class distribution and ensured objective model evaluation.

6. TensorFlow Data Pipeline

The tf.data API was used to build an optimized input pipeline. Each patch was loaded, decoded, resized to 160×160, normalized to [0,1], and augmented using random flips, rotations, and zoom. Data was batched and prefetched to maximize GPU/CPU training throughput.

7. CNN Model Architecture

A lightweight convolutional neural network was implemented. The architecture included three Conv2D + MaxPooling blocks, followed by GlobalAveragePooling, a dense layer with dropout, and a sigmoid output layer. The model contained approximately 101,569 trainable parameters, making it efficient for real-time or large-scale inference.

8. Training Configuration

Training was performed for 5 epochs using the Adam optimizer and binary cross-entropy loss. Data augmentation improved generalization by exposing the model to varied pavement textures and orientations. Batch size was set to 32.

9. Evaluation Results

The trained model achieved 86.13% accuracy on the test dataset. Distress recall reached 98%, indicating that the classifier rarely missed true distressed patches. Healthy recall was 27%, reflecting a conservative bias where uncertain cases were classified as distressed. The confusion matrix was: True Healthy → Pred Healthy: 97 True Healthy → Pred Distress: 267 True Distress → Pred Healthy: 32 True Distress → Pred Distress: 1760

10. Strengths and Limitations

The system demonstrates strong performance in detecting distressed pavement, with excellent sensitivity to cracks. The modular design and patch-based approach support scalability for broader digital-twin or GIS-driven applications. Limitations include dataset imbalance, lower healthy recall, and limited contextual information from 160×160 patches.

11. Future Improvements

Potential enhancements include class balancing using weighted loss or oversampling, increasing patch sizes for more context, multi-class labeling of different distress types, and implementing sliding-window full-image inference to produce crack probability heatmaps or geospatial layers.

12. Conclusion

The project delivers a fully functional, end-to-end deep learning pipeline for UAV-based pavement distress detection. The workflow is reproducible, computationally efficient, and adaptable for real-world infrastructure inspection tasks. It forms a robust foundation for more advanced geospatial analysis and deep learning applications.