Pothole Detection Using YOLOv8x and Image Processing

Group Members:
Ajay Deep Rawat (202211001)
Tejas Pakhale (202211061)
Rahul Gupta (202211069)

Under the guidance of:
Dr. Deepika Gupta

May 4, 2025

Introduction

- **Objective:** Leverage the YOLOv8x deep learning model for efficient pothole detection.
- **Image Processing Techniques:** Incorporate advanced methods to enhance the dataset for improved model accuracy.
- **Process Overview:**
  - Preprocessing: Clean and prepare the data for model input.
  - Annotation Transformation: Convert and standardize labels to fit the model requirements.
  - Data Augmentation: Apply transformations like rotation, flipping, and scaling to increase dataset diversity.

Methodology Overview

- **Data Preprocessing:** Preparing and cleaning raw images for optimal model input.
- **Image Enhancement:** Enhancing image quality for better feature extraction.
- **Perspective Transformation:** Adjusting image perspectives for accurate object detection.
- **Bounding Box Adjustment:** Fine-tuning the boundaries around objects for precise localization.
- **Data Augmentation:** Applying transformations (rotation, scaling, etc.) to expand dataset diversity.
- **Model Training and Inference:** Training the YOLOv8x model and using it for real-time pothole detection.

Project Workflow

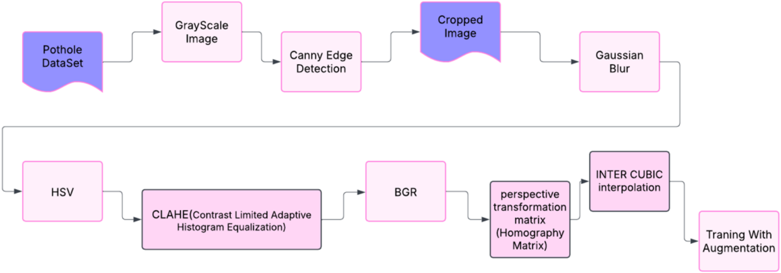


Figure 1: Workflow Diagram

Data Preprocessing

- **Dataset Splitting:**
  - Training Set: 70% of the data used for model training.
  - Validation Set: 15% for model validation during training.
  - Test Set: 15% reserved for final model evaluation.
- **Grayscale Conversion:** Convert images to grayscale to enhance edge detection.
- **Canny Edge Detection:**
  - **Lower Threshold:** 50 – Pixels with gradients below this are rejected.
  - **Upper Threshold:** 150 – Pixels with gradients above this are accepted.
  - **Edge Connectivity:** Pixels between the thresholds are considered edges if connected to stronger edges (above 150).
- **Bounding Box Extraction:** Extract bounding boxes from annotation files for object localization.
- **YOLO to Pixel Conversion:** Convert YOLO's normalized bounding box coordinates to pixel values for precise localization.

Figure 1: Original Image

Figure 2: Grayscale Image

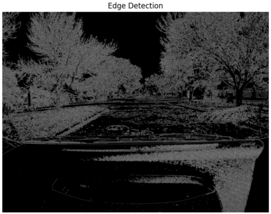


Figure 3: Preprocessed Image

ROI Extraction and Cropping

- **ROI Selection:**
  - Identify the Region of Interest (ROI) around detected potholes for focused processing.
- **Margin Addition:**
  - Add a 20% margin to both the width and height of the ROI to ensure better object coverage.
- **Image Cropping:**
  - Extract the ROI from the original image for further analysis and processing.
- **Bounding Box Adjustment:**
  - Recalculate bounding boxes to match the cropped image and maintain accurate object localization.
- **Image Annotation:**
  - Draw bounding boxes and polylines on the cropped image for visualization and annotation.
- **Cropped Image Saving:**
  - Save the processed and cropped images for use in further model training.

ROI Extraction and Cropping

Figure 1: Bounding Boxes

Figure 2: Cropped Output

Figure 3: ROI Detected

Image Enhancement Techniques

- **Gaussian Blur (5x5 Kernel)**:
  - Smoothens the image and reduces noise.
  - Kernel size of 5x5 applied to each pixel neighborhood.
  - Standard deviation (σ) automatically calculated based on kernel size.
- **Color Space Conversion:**
  - Converts image from BGR to HSV (Hue, Saturation, Value).
  - HSV representation enhances feature extraction by separating chromatic and intensity components.
- **Histogram Equalization (CLAHE on V Channel)**:
  - **Contrast Enhancement**:
    - Applied to the Value (V) channel of the HSV image.
    - `clipLimit` = 2.0 to control the level of contrast enhancement.
    - `tileGridSize` = (8,8) to enhance contrast locally in regions of the image.
- **Revert to BGR:**
  - Merge the enhanced V channel back into the HSV image.
  - Convert the image from HSV back to BGR color space for final output.

Image Enhancement Techniques

Figure 1: Original Image

Figure 2: Noise Reduced Image

Figure 3: HSV Converted

Figure 4: Histogram Equalized Image

Figure 5: HSV to BGR image

Perspective Transformation and Warping

- **Homography Calculation:**
  - Computes the transformation matrix to correct the perspective of the image.
  - Utilizes point correspondences to derive the homography matrix.
- **Warp Perspective:**
  - Applies the homography transformation to generate a rectified image with corrected perspective.
- **Bicubic Interpolation:**
  - Enhances image quality during the warping process.
  - Provides smoother and more accurate pixel interpolation compared to bilinear interpolation.
- **ROI Adjustment Coordinate Transformation:**
  - Adjusts the Region of Interest (ROI) coordinates after the perspective transformation.
  - Ensures bounding box alignment with the warped image.

Perspective Transformation and Warping

Figure 1: Warped Image

Figure 2: Interpolated Image

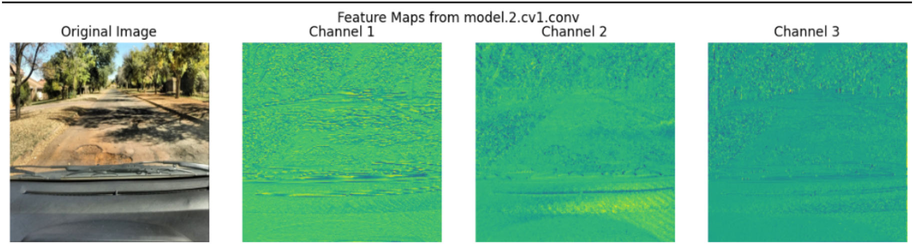
Dataset Augmentation for Model Training

- **HSV Adjustments:**
  - Modifies the Hue ($hsv\_h = 0.015$) to introduce slight color shifts.
  - Alters the Saturation ($hsv\_s = 0.7$) to vary the intensity of colors.
  - Adjusts the Value ($hsv\_v = 0.4$) to simulate different lighting conditions.
- **Simulating Lighting and Weather Conditions:**
  - Enhances the dataset by creating variations in color properties.
  - Mimics real-world environmental changes to improve model robustness.

Model Training and Inference

- **YOLO Model Training:**
  - Train the YOLOv8x model using the processed and augmented dataset.
  - Optimize the model for accurate pothole detection in various environments.
- **Inference on Test Images:**
  - Use the trained YOLO model to perform inference on unseen test images.
  - Detect and classify potholes based on learned patterns and features.
- **Bounding Box Drawing:**
  - Draw green bounding boxes around detected potholes.
  - Display confidence scores along with the bounding boxes to indicate the certainty of detection.

Model Training and Inference



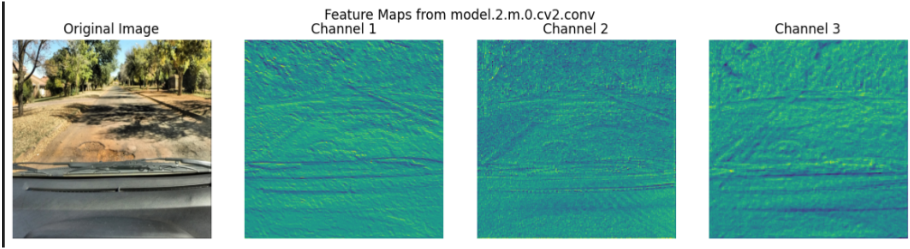


Figure: Example of detected potholes with bounding boxes

Metrics

Phase	Images	P (Precision)	R (Recall)	mAP@0.5 — mAP@0.5:0.95
Method Given in Paper	187	0.716	0.569	0.663 — 0.353
Our Method	101	0.768	0.660	0.772 — 0.517

Table: Comparison of object detection metrics

Future Scope

Figure 1: Original Image

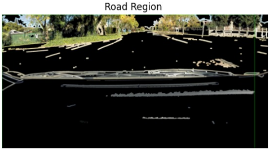


Figure 2: Road Region

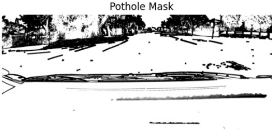


Figure 3: Pothole Mask