# Road Anomalies Detection Project Documentation

## 1. Introduction

This document outlines the scope, objectives, methodology, and expected outcomes of the Road Anomalies Detection Project. The primary goal of this project is to develop a robust and efficient system for identifying and reporting various types of road anomalies, such as potholes, cracks, uneven surfaces, and missing road signs. Early detection of these anomalies is crucial for timely maintenance, enhancing road safety, and improving overall infrastructure management.

## 2. Problem Statement

Road anomalies pose significant risks to drivers, pedestrians, and vehicles, leading to accidents, vehicle damage, and uncomfortable journeys.

Traditional methods of anomaly detection often rely on manual inspections, which are time-consuming, labor-intensive, and prone to human error.

There is a critical need for an automated, scalable, and accurate system that can continuously monitor road conditions and provide real-time or near real-time anomaly alerts.

## 3. Objectives

The main objectives of this project are:

To design and implement a system capable of automatically detecting various road anomalies.

To achieve high accuracy in identifying different types of anomalies (e.g., potholes, cracks, bumps).

To provide a mechanism for reporting detected anomalies, including their location and type.

To develop a scalable solution that can be deployed across large road networks.

To contribute to improved road safety and reduced maintenance costs.

## 4. Methodology/Approach

**4.1. Data Acquisition and Pre-processing:**

**Video Capture:** Videos containing geographic location data are captured using the GPS MAP CAMERA LITE application. This ensures that each captured video is associated with precise real-world coordinates.

**Working Environment:** The captured videos are then uploaded to a Google Colab working directory, providing a cloud-based environment for processing.

**Frame Extraction:** A Python script utilizing the opencv and os libraries is employed to extract individual frames from the videos. This script is designed to extract frames at approximately 3-meter intervals along the road, ensuring a consistent and representative sampling of the road surface.([Frames Extraction](Frames Extraction.docx))

**Initial Dataset:** These extracted frames form the raw dataset for further processing.

**4.2. Data Annotation and Labelling:**

**Annotation Process:** The extracted frames are meticulously annotated to create a comprehensive dataset for model training.

This involves drawing various polygon shapes around different road anomalies (e.g., potholes, cracks, manholes, road signs, uneven surfaces) to precisely delineate their boundaries.

Annotation Tool: For the annotation and labelling task, Roboflow is utilized. ([Methods](Labelling Methods.docx))

Roboflow provides a streamlined platform for managing datasets, performing annotations, and preparing data for machine learning models. (Further details on Roboflow are provided in Section 8).

**Training Dataset:** Through this process, a training dataset of approximately 1300 annotated images is generated, serving as the foundation for training the anomaly detection model.

**4.3. Anomaly Detection Algorithms (Model Training):**

**Machine Learning Application:** The annotated dataset of 1300 images is used for training the anomaly detection model.

This involves applying machine learning principles and Python programming to develop a robust model capable of identifying road anomalies.

**Ultralytics** is an AI/ML company that specializes in computer vision and deep learning, particularly **real-time object detection**, **segmentation**, **classification**, and **tracking**.

They are best known for developing: **YOLOv5** (2020) **YOLOv8** (2023) **Ultralytics Hub** – a cloud platform for training and deploying vision models

**Model Performance:** The developed model has demonstrated effective performance in detecting various road anomalies, indicating its readiness for practical application.

It gives the 90% accuracy in anomalies detection.

**GPS Integration:** Accurately mapping detected anomalies to geographical coordinates.

.

## 5. Key Features/Components

**Data Collection Unit:** GPS MAP CAMERA LITE application for video capture with geo-location.

**Processing Module:** Google Colab environment for video processing and frame extraction.

**Annotation Tool:** Roboflow for efficient dataset annotation and management.

**Machine Learning Model:** **YOLOv5** is a family of object detection models that detect objects in images and videos in real time.

YOLOv8 is a state-of-the-art, **PyTorch-based** model that supports:

**Object Detection**

**Instance Segmentation**

**Image Classification**

## 6. Expected Outcomes

A functional prototype or a deployed system for automated road anomaly detection.

Reduced manual inspection efforts and associated costs.

Improved accuracy and consistency in anomaly identification.

Faster response times for road maintenance, leading to safer roads.

A comprehensive database of road anomaly data for analysis and planning.

.

## 7. Future Enhancements

Predictive maintenance capabilities based on anomaly growth patterns.

Crowdsourcing anomaly data from public users via mobile applications.

Advanced anomaly severity assessment and prioritization.

.

## 8. Data Management and Reporting Structure

### 8.1. Folder Structure

To ensure organized data management, the project will adhere to the following folder structure:

Raw Images: This folder will store all original, unprocessed image data collected from various sources.

Training Images: This folder will contain images specifically prepared and annotated for training the machine learning models.

Prediction Output: This folder will store the results of the anomaly detection process, including images with detected anomalies highlighted and associated metadata.

Within these main folders, images will be further organized by location and defect category:

For each distinct location where data is collected, a dedicated subfolder will be created.

Inside each location folder, additional subfolders will be created for each identified defect category (e.g., "Pothole", "Crack", "Damage Road","lane divider", "edge line").

.

### 8.2. Final Report Generation

A crucial output of the project will be a summarized report in CSV format, providing an overview of detected anomalies. This report will include the following details:

Location: The geographical area where anomalies were detected.

Defect Category Count: The total number of anomalies identified for each specific defect category within that location.([Count](Summary Count.csv))

This CSV file will serve as the primary final report, offering a concise and actionable summary for road authorities and maintenance teams.

### 9. Flowchart for Roboflow Working

A[Start] --> B(Upload Raw Images to Roboflow)

↓

B --> C(Annotate Images with Polygons/Labels)

↓

C --> D(Apply Preprocessing Steps)

↓

D --> E(Apply Data Augmentation)

↓

E --> F(Generate New Dataset Version)

↓

F --> G(Export Dataset in Desired Format)

↓

G --> H(End)

# 10. **Tools and Technologies Used**

Programming Language: Python

Deep Learning: Roboflow, YOLOv5, YOLOv8

Video Processing: OpenCV, Os

Data Annotation: Roboflow

Workflow Automation: Jupyter Notebook, Google Colab

## 11.Overall Project Flowchart

A[Start] --> B(Capture Videos with Google Map Cam)

↓

B --> C(Upload Videos to Google Colab)

↓

C --> D(Extract Frames at 3m Distance using Python/OpenCV)

↓

D --> E(Upload Frames to Roboflow)

↓

E --> F(Annotate Frames in Roboflow)

↓

F --> G(Preprocess & Augment Data in Roboflow)

↓

G --> H(Export Annotated Dataset from Roboflow)

↓

H --> I(Train Anomaly Detection Model using ML/Python)

↓

I --> J{Model Performs Well?}

↓

J -- No --> G

↓

J -- Yes --> K(Detect Anomalies in New Data)

↓

K --> L(Organize Prediction Output by Location/Defect)

↓

L --> M(Generate Final CSV Report)

↓

M --> N(End)