Technical Summary – Road Surface Anomaly Detection using an AI-Driven CPS

Introduction

This project actually started from a very real and everyday problem: our roads.
Anyone who travels frequently — whether by bike, car, bus or even on foot — knows how common potholes, cracks, and uneven surfaces are. People complain about them all the time, and yet repairs take ages because the entire system still depends heavily on manual inspection, on-site visits, or someone raising a complaint.

We wanted to build something practical, realistic, and low-cost — not a fancy research-lab-only solution. So the idea was simple:

Use just a camera (like the one on a normal phone or a dashcam), apply AI to the images, automatically detect road damage, and show exactly where the damage exists on a map.

This approach fits perfectly under a **Cyber-Physical System (CPS)** because we're combining the physical world (camera, GPS, roads) with the cyber world (AI model, predictions, dashboard).

1. Understanding the Problem (Why this project matters)

Road maintenance teams usually depend on three things:

1. Manual surveying by workers
2. Occasional monitoring vehicles
3. Complaints from the public

All of these are slow, inconsistent, and expensive. Cities have long road networks, and it's impossible to manually cover everything regularly.

We asked ourselves:
"What if a simple mobile phone camera could do the job automatically?"

With AI getting better and more accessible, you don't need special sensors or expensive equipment. A camera + GPS + deep learning is enough to build a basic working system.

So our CPS idea became:

- Use any camera (bike, car, phone)
- Capture road images while driving
- AI model detects potholes/cracks
- If images have GPS data, plot the location on a map
- Generate a quick dashboard so authorities can see problem areas visually

The goal was never to create a perfect research-grade system — the goal was to create something **practical, low-cost, and immediately useful**.

2. Dataset and Pre-Processing

For road damage detection, the images came from two sources:

A. Existing open datasets

Examples include:

- India Road Damage Dataset (IRDD)
- Japan Road Damage Dataset
- Pothole-600
- Other publicly available road anomaly sets

These datasets contain potholes, cracks, patches, and uneven surfaces.

B. Self-collected images

Using a mobile camera while traveling.
This makes the dataset more relevant to typical Indian road conditions.

Annotation & Labeling

The images were labeled using tools like **LabelImg** or **Roboflow**, marking:

- potholes
- cracks
- uneven surfaces
- road patches

Annotations were exported in **YOLO text format**, which is simple:

class x_center y_center width height

Pre-processing

Before training, images went through:

- Resizing to 640×640
- Normalization
- Random horizontal flips
- Random brightness / contrast changes
- Light motion blur (simulating movement)
- Optional rain/fog overlay augmentation

The idea behind augmenting is simple:
AI models learn better when you show them many versions of the same kind of image.

This helps make the model more robust to real-world lighting and weather.

3. Model Architecture, Training & Evaluation

We chose **YOLOv8** for the model. Reasons:

- Fast
- Accurate
- Easy to train
- Lightweight enough to run even on normal systems
- Popular and well-supported

The model takes an input image and returns:

- Bounding boxes
- Class labels (pothole, crack, etc.)
- Confidence scores

Training Setup

The training was done inside the notebook
"Road_Surface_Anomaly_Detection_using_AI_Driven_CPS.ipynb".

Key training choices:

- Input size: 640×640
- Batch size: 16
- Epochs: 50–100 (depending on dataset portion)
- Optimizer: Adam or SGD
- Loss functions: bounding box loss + objectness loss + classification loss

Dataset split:

- 70% training
- 20% validation
- 10% testing

Evaluation Metrics

We measured:

- **mAP50** (mean average precision at IoU 0.5)

- **mAP50–95**
- **Precision**
- **Recall**
- **F1-Score**

The model achieved **stable and reliable performance**, enough to be used in a practical prototype.

Even though road images vary a lot (lighting, traffic, angles), YOLOv8 handled them quite well.

4. CPS Dashboard & Visualization

Detecting potholes is only half the job.
The other half is making useful **visual outputs**.

So we built a lightweight dashboard using:

- Annotated images (with bounding boxes)
- GPS extraction (from image EXIF data, if available)
- A map showing all anomaly locations using **Folium**
- A detections summary table

This makes it extremely easy for a human reviewer (like a municipal engineer) to:

- See which roads have the most damage
- View exact coordinates
- Check annotated visuals
- Prioritize repair areas

The dashboard is generated completely inside the notebook, with **no need for servers or ngrok**. A simple HTML file is created which can be opened anywhere.

5. Deployment Thoughts

Since the hackathon required a simple deployable idea but not a full cloud deployment, we kept things light.

The system supports:

- Local inference
- Local dashboard HTML output
- Optionally, a Streamlit app
- Optionally, a FastAPI backend

But for the purpose of this project submission:

- Everything runs inside the notebook
- Outputs are generated automatically
- No extra setup is required

This keeps the system simple, beginner-friendly, and easy to demonstrate.

Later, if required, it can be converted into:

- A mobile app
- A dashcam-based automatic reporting system
- A cloud dashboard for municipal authorities

This flexibility is the strength of a CPS approach.

6. Final Remarks and Summary

This project shows how AI can be used meaningfully for civic improvement without needing heavy budgets or complex equipment.

What we achieved:

- A working AI model that detects potholes and cracks
- A simple dataset pipeline with annotations and augmentation
- A YOLOv8 training setup inside a single notebook
- A visualization dashboard showing detections + map locations
- A CPS-style structure combining sensors, AI, and mapping
- A practical solution that can be scaled, improved, or deployed anywhere

Even though it's a prototype, the system already proves that:

A basic camera + smart AI = a powerful low-cost tool for road quality monitoring.

All code, outputs, and the final notebook are prepared and ready for GitHub submission and final evaluation.