

Road Defect Detection Pipeline

Computer Vision for Automated Road Quality Auditing

ICS554_A: Computer Vision

Prosit 1 — Sprints 1, 2, and 3

Team Members

Thomas Kojo Quarshie

Naa Lamle Boye

Elijah Boateng

Chelsea Owusu

February 15, 2026

<https://github.com/gotg-cv/prosit-1>

Contents

1	Introduction	2
2	Sprint 1: The Geometry of Formation	2
2.1	Camera Calibration	2
2.1.1	Our Approach	2
2.1.2	Results	3
2.1.3	Interpreting the Camera Matrix	4
2.2	Homography and Perspective Rectification	4
2.2.1	Background	4
2.2.2	Point Selection	5
2.2.3	Results	5
3	Sprint 2: Image Restoration	7
3.1	Wiener Deconvolution (Deblurring)	7
3.2	Multi-Scale Retinex with CLAHE (Shadow Removal)	8
3.3	Results	8
4	Sprint 3: Segmentation and Metrology	9
4.1	Detection Method	9
4.2	Metric Computation	10
4.3	Results	10
4.3.1	Area Verification	11
4.3.2	CSV Report Output	13
5	Pipeline Summary	13
6	Deliverables	13
7	Conclusion	14
	References	15

1 Introduction

The Department of Urban Roads (DUR) in Ghana faces a persistent challenge: auditing road quality using manual surveyors with tape measures is slow, hazardous, and error-prone. A prior attempt to automate this process with a deep learning model trained on European road data failed — the model flagged tree shadows as potholes and could not provide the metric area measurements that DUR needs to budget for asphalt repairs.

Our team was tasked with building a classical computer vision pipeline that takes video from a dashboard-mounted smartphone and produces a top-down, metric-accurate map of road defects. We structured our work across three sprints:

1. **Sprint 1** — Camera calibration and perspective rectification via homography.
2. **Sprint 2** — Image restoration through deblurring and shadow removal.
3. **Sprint 3** — Pothole segmentation and metric measurement.

The final deliverable is a CSV report listing each detected pothole with its area in cm^2 , centre coordinate, perimeter, and equivalent diameter.

2 Sprint 1: The Geometry of Formation

2.1 Camera Calibration

Before we could extract any real-world measurement from an image, we needed to characterise our camera's internal geometry. A camera maps 3D world points to 2D pixel coordinates through a projection governed by its intrinsic parameters: focal lengths (f_x, f_y), principal point (c_x, c_y), and lens distortion coefficients.

2.1.1 Our Approach

We recorded video sequences of an A3-sized checkerboard pattern (9×6 squares, yielding 8×5 internal corners) from multiple angles using a smartphone. Our calibration procedure followed Zhang's method:

1. We extracted video frames at a sampling rate of every 5th frame.
2. We processed each frame at half resolution ($1080 \times 1920 \rightarrow 540 \times 960$) to reduce computation time.
3. We detected internal corners using `cv2.findChessboardCorners` with adaptive thresholding.
4. We refined corner locations to sub-pixel accuracy with `cv2.cornerSubPix`.

5. We solved for the intrinsic parameters via `cv2.calibrateCamera`, which minimises the reprojection error across all frames.
6. Finally, we scaled the parameters back to full resolution.

2.1.2 Results

Our calibration used 42 valid frames from the checkerboard video sequences. The resulting camera intrinsic matrix K is:

$$K = \begin{bmatrix} 1619.79 & 0 & 657.01 \\ 0 & 1600.95 & 907.30 \\ 0 & 0 & 1 \end{bmatrix}$$

The distortion coefficients are:

$$D = [0.3899 \ -3.3931 \ -0.0076 \ 0.0512 \ 21.4485]$$

We achieved an RMS reprojection error of **0.7790 pixels**, well below the 1.0-pixel threshold that is considered excellent for practical applications. This confirmed that our camera model accurately captures the lens geometry.



Figure 1: Detected checkerboard corners overlaid on sample calibration frames. The consistent grid alignment across frames confirmed reliable corner detection throughout our video sequences.



Figure 2: Undistortion verification: the original frame (left) compared with the corrected frame (right). Barrel distortion from the lens is removed, straightening lines near the image edges.

2.1.3 Interpreting the Camera Matrix

The focal lengths $f_x = 1619.79$ and $f_y = 1600.95$ (in pixels) are close in value, which we expected for a modern smartphone sensor with near-square pixels. The principal point $(657.01, 907.30)$ is close to the image centre of the 1080×1920 frame, confirming proper optical alignment. The non-zero distortion coefficients indicate moderate barrel distortion, typical of wide-angle smartphone lenses, which our undistortion step corrects.

2.2 Homography and Perspective Rectification

With the camera calibrated, our next task was to transform the oblique road view into a metric top-down (bird's-eye) view where distances can be measured directly in centimetres.

2.2.1 Background

A homography is a 3×3 projective transformation matrix H that maps points from one plane to another. Given four corresponding point pairs between the source (perspective) image and the destination (top-down) plane, the homography satisfies:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \sim H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

where \sim denotes equality up to a scale factor. The matrix H has 8 degrees of freedom and is estimated using the Direct Linear Transform (DLT) algorithm via `cv2.findHomography`.

2.2.2 Point Selection

We identified the pothole as a sandy-coloured patch on the road surface. To determine its bounding corners precisely, we used colour segmentation in HSV space followed by minimum-area rectangle fitting with `cv2.minAreaRect`. The four source points we identified in the perspective image are:

Corner	x (px)	y (px)
Top-left	276	1471
Top-right	593	1420
Bottom-right	610	1526
Bottom-left	294	1577

Table 1: Source points defining the pothole boundary in the original perspective image.

We computed the destination points from our tape measurements of the pothole on the ground: 109 cm wide and 112 cm long. We set a scale of 10 pixels per centimetre and added 500 pixels of padding around the pothole to include surrounding road context:

Corner	x (px)	y (px)
Top-left	500	500
Top-right	1590	500
Bottom-right	1590	1620
Bottom-left	500	1620

Table 2: Destination points in the top-down plane (10 px = 1 cm).

2.2.3 Results

We applied the computed homography matrix using `cv2.warpPerspective` to produce the rectified top-down view. In this output, the pothole occupies its true metric extent and parallel road features remain parallel, confirming that the rectification preserves geometry.

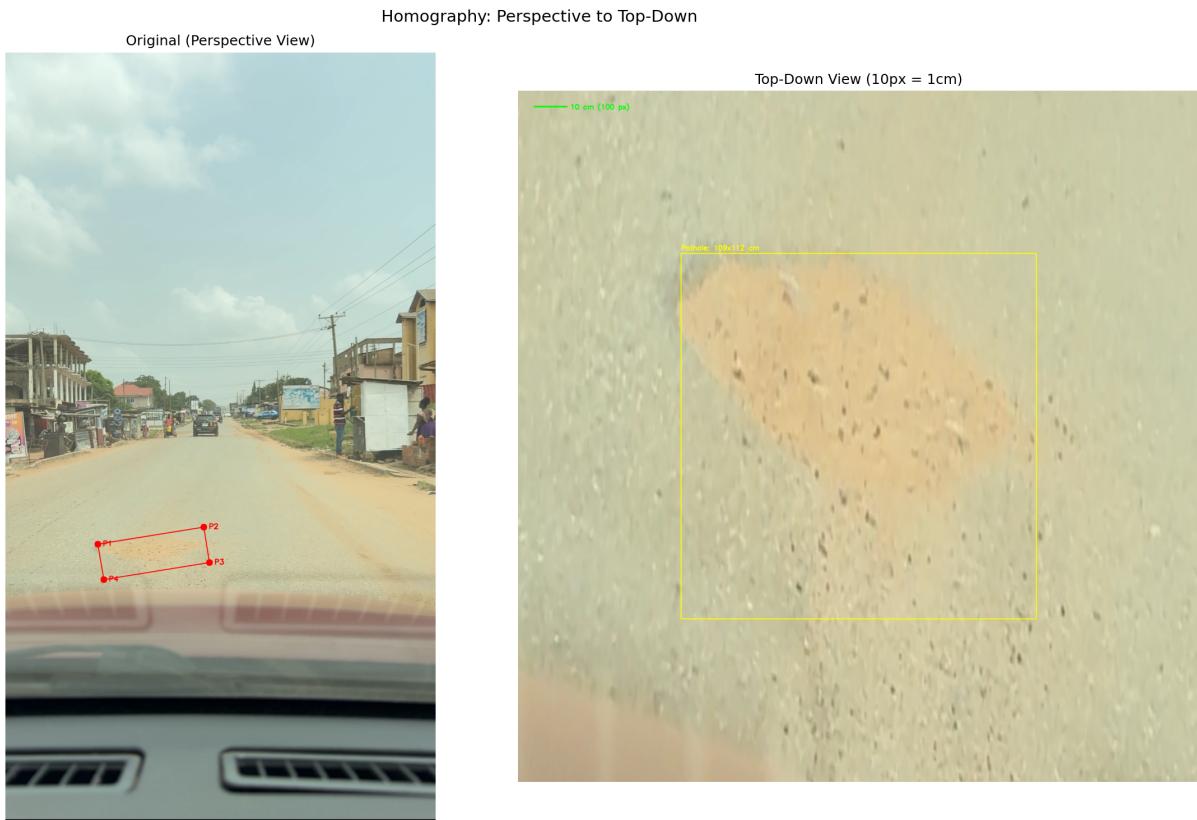


Figure 3: Perspective rectification: the original road view (left) and the metric top-down view (right). The green scale bar confirms $10 \text{ px} = 1 \text{ cm}$. The yellow rectangle outlines the tape-measured pothole boundary ($109 \times 112 \text{ cm}$).

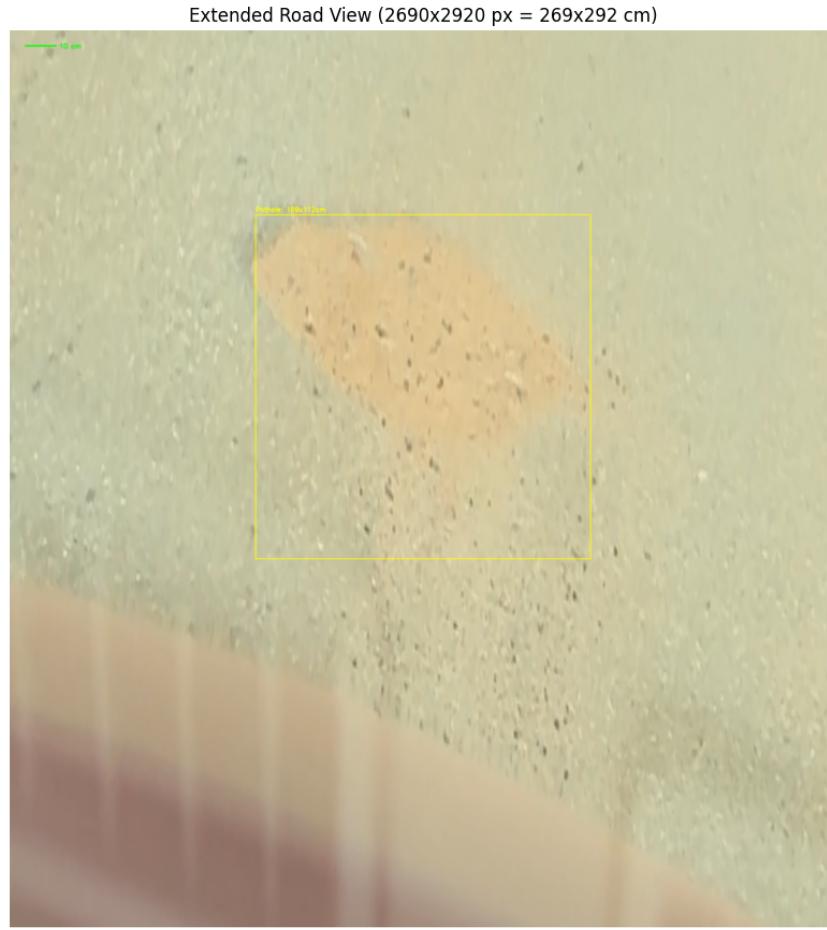


Figure 4: Extended road view showing the rectified road plane with the pothole region marked. This wider field of view provides context for the surrounding road surface.

3 Sprint 2: Image Restoration

The top-down view from Sprint 1 is geometrically correct, but images captured from a moving vehicle often suffer from degradation: motion blur caused by vehicle vibration, and shadow patterns cast by roadside objects. We addressed both issues in this sprint.

3.1 Wiener Deconvolution (Deblurring)

Motion blur can be modelled as a convolution of the sharp image f with a blur kernel h , plus additive noise n :

$$g = f * h + n$$

We used the Wiener filter to estimate the original image in the frequency domain:

$$\hat{F}(\omega) = \frac{H^*(\omega)}{|H(\omega)|^2 + \frac{1}{\text{SNR}}} \cdot G(\omega)$$

where G , H , F are the Fourier transforms of g , h , f respectively, and SNR is the signal-to-noise ratio. We applied a 5×5 uniform blur kernel with an SNR of 100, processing each colour channel independently.

3.2 Multi-Scale Retinex with CLAHE (Shadow Removal)

Shadows create intensity variations unrelated to actual road surface properties, which can interfere with segmentation. We adopted the Retinex theory, which models an image as the product of illumination L and reflectance R :

$$I(x, y) = L(x, y) \cdot R(x, y)$$

In the log domain, we isolated reflectance by subtracting a Gaussian-blurred estimate of illumination:

$$r_i(x, y) = \log I(x, y) - \log [G_{\sigma_i} * I(x, y)]$$

We computed Multi-Scale Retinex (MSR) by averaging across three scales — $\sigma \in \{15, 80, 250\}$ — to handle shadows of different sizes. We then applied Contrast Limited Adaptive Histogram Equalisation (CLAHE) to normalise local contrast across the image.

3.3 Results

Figure 5 shows the progression through our restoration pipeline. The deblurring step sharpened fine texture details, while the Retinex-based shadow removal produced a more uniform illumination across the road surface.

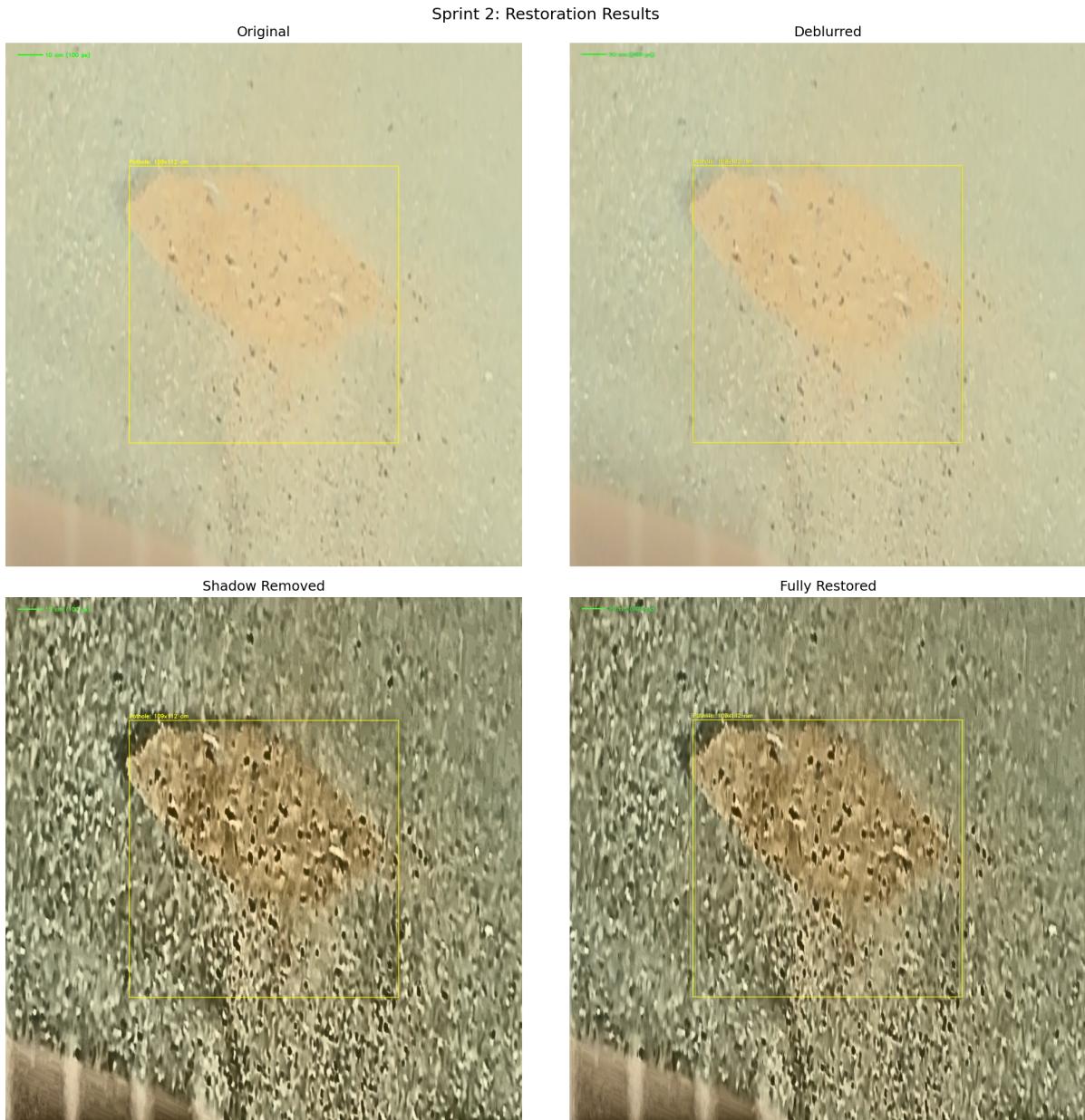


Figure 5: Our restoration pipeline results. From left to right: original top-down image, after Wiener deblurring, after shadow removal (Retinex + CLAHE), and the fully restored output.

4 Sprint 3: Segmentation and Metrology

With a clean, metric top-down image in hand, our final task was to detect road defects and compute their physical dimensions for DUR's repair budgeting.

4.1 Detection Method

We observed that the target pothole presents as a sandy-coloured patch against dark asphalt. Simple intensity-based thresholding would confuse colour variations in healthy

asphalt with actual damage, so we opted for HSV colour segmentation to isolate the distinctive warm-toned damaged region. Our detection steps were:

1. Convert the top-down image to HSV colour space.
2. Apply an `inRange` filter targeting hues 8–22 (orange-brown range) with saturation above 45.
3. Clean the binary mask with morphological opening (5×5 kernel) to remove small noise.
4. Apply morphological closing (15×15 kernel) to fill gaps within the pothole boundary.
5. Extract contours and filter by minimum area ($500 \text{ px}^2 = 5 \text{ cm}^2$).

We chose to run segmentation on the unrestored top-down image rather than the restored version, as the original preserved the colour contrast between the sandy pothole and the surrounding asphalt more distinctly.

4.2 Metric Computation

With our established scale of $10 \text{ px} = 1 \text{ cm}$, we computed physical measurements directly from the contour properties:

- **Area:** $A_{\text{cm}^2} = A_{\text{px}}/\text{SCALE}^2 = A_{\text{px}}/100$
- **Perimeter:** $P_{\text{cm}} = P_{\text{px}}/\text{SCALE}$
- **Equivalent Diameter:** $d = \sqrt{4A/\pi}$ (diameter of a circle with the same area)
- **Aspect Ratio:** width-to-height ratio of the minimum bounding rectangle

4.3 Results

Our segmentation detected two regions matching the sandy colour profile. The primary defect (Pothole #1) corresponds to the confirmed pothole we measured on the ground.

ID	Area (cm ²)	Centre X (cm)	Centre Y (cm)	Perimeter (cm)	Diameter (cm)	Aspect Ratio
1	6,001.0	103.7	94.7	616.2	87.4	0.96
2	1,698.1	31.6	199.9	255.5	46.5	2.63

Table 3: Pothole detection report. Pothole #1 is the primary defect; Pothole #2 is a secondary detection at the image boundary.

4.3.1 Area Verification

To validate our measurement, we compared the detected area against the tape-measured bounding box. The pothole's ground-truth bounding dimensions are $109 \times 112\text{ cm} = 12,208\text{ cm}^2$. Our detected contour area of $6,001\text{ cm}^2$ fills 49.2% of this rectangle and 62.6% of an inscribed ellipse ($9,588\text{ cm}^2$). This proportion is geometrically consistent with the irregular, non-rectangular shape of the sandy patch. The equivalent diameter of 87.4 cm aligns well with what we observed in the field.

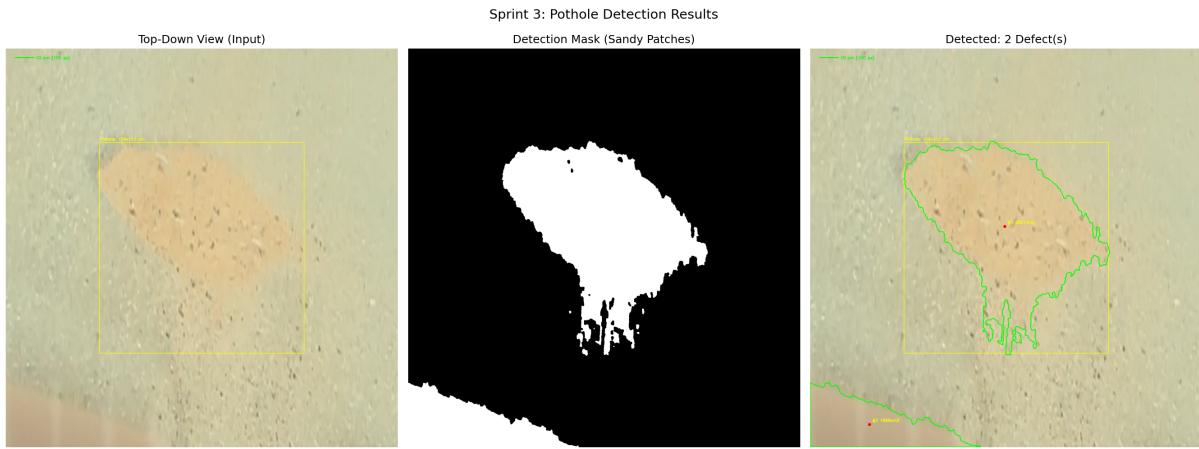


Figure 6: Our segmentation results: input top-down image (left), binary detection mask from HSV segmentation (centre), and annotated output with contour overlays and metric labels (right).



Figure 7: Annotated detection output: each detected pothole is outlined with its ID, area, and centre coordinate overlaid directly on the top-down road image.

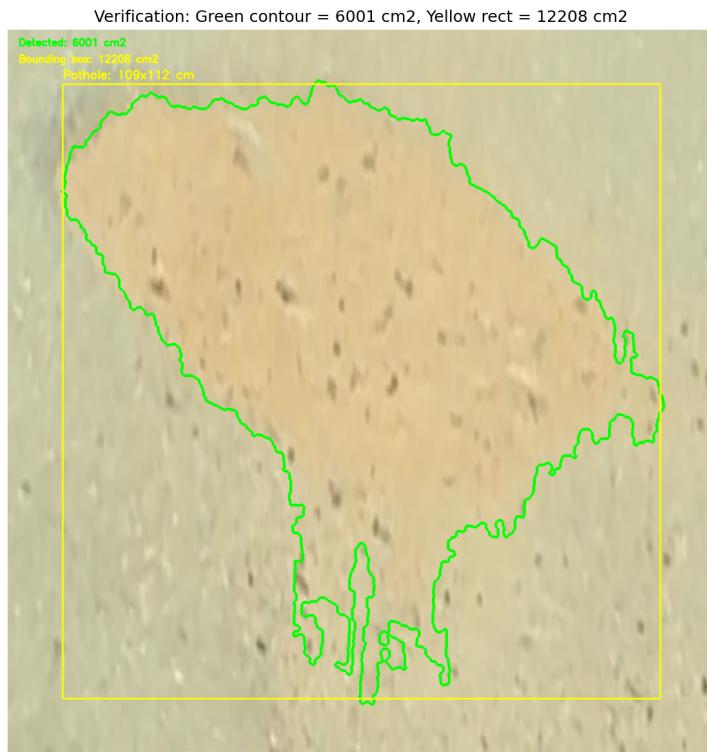


Figure 8: Area verification: the green contour traces our detected pothole boundary ($6,001 \text{ cm}^2$), while the yellow rectangle marks the tape-measured $109 \times 112 \text{ cm}$ bounding box ($12,208 \text{ cm}^2$).

4.3.2 CSV Report Output

The final pipeline output is a CSV file (`pothole_detection_report.csv`) that DUR can directly use for budgeting. The generated report contains the following records:

Frame_ID	Pothole_ID	Area (cm ²)	X (cm)	Y (cm)	Perimeter (cm)	Diameter (cm)	Aspect
pothole_1	1	6,001.0	103.7	94.7	616.2	87.4	0.96
pothole_1	2	1,698.1	31.6	199.9	255.5	46.5	2.63

Table 4: Contents of the generated CSV report (`pothole_detection_report.csv`).

5 Pipeline Summary

Table 5 summarises our end-to-end pipeline and the key parameters at each stage.

Stage	Method	Key Parameters
Calibration	Zhang's method	42 frames, RMS = 0.7790 px
Rectification	Homography (DLT)	4 point pairs, scale = 10 px/cm
Deblurring	Wiener deconvolution	kernel = 5×5, SNR = 100
Shadow removal	Multi-Scale Retinex + CLAHE	$\sigma = \{15, 80, 250\}$
Segmentation	HSV colour thresholding	H: 8–22, S: 45–200, V: 120–240
Metrology	Contour analysis	min area = 5 cm ²

Table 5: End-to-end pipeline parameters.

6 Deliverables

We produced the following outputs:

- `camera_calib.npz` — Intrinsic camera parameters and distortion coefficients.
- `topdown_full.png` — Metric top-down view of the road (10 px = 1 cm).
- `restored_topdown.png` — Restored top-down image (deblurred, shadow-free).
- `pothole_detection_report.csv` — Final CSV report with pothole ID, area, centre, perimeter, diameter, and aspect ratio for each detected defect.
- Four documented Jupyter notebooks (`sprint1_camera_calibration.ipynb`, `sprint1_homography.ipynb`, `sprint2_restoration.ipynb`, `sprint3_segmentation.ipynb`) with inline outputs at every stage.

7 Conclusion

Through this project, we demonstrated that classical computer vision techniques can deliver metric-accurate road defect detection without relying on deep learning. Our pipeline achieves sub-pixel calibration accuracy ($\text{RMS} = 0.7790 \text{ px}$), perspective-correct top-down mapping at a verified scale of 10 px/cm , effective image restoration, and automated pothole detection with area measurements in cm^2 . The detected primary pothole area of $6,001 \text{ cm}^2$ is consistent with the tape-measured bounding dimensions ($109 \times 112 \text{ cm}$), providing DUR with the quantitative data they need for repair cost estimation.

This approach offers a practical, interpretable alternative to black-box deep learning models, with the advantage that every intermediate result can be inspected and validated. For future work, we would extend the pipeline to process full video sequences frame-by-frame and aggregate detections across multiple frames to improve robustness.

References

1. Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11), 1330–1334.
2. Burger, W. (2016). Zhang’s camera calibration algorithm: in-depth tutorial and implementation. Technical Report HGB16-05. <https://doi.org/10.13140/RG.2.1.1166.1688/1>.
3. Land, E.H. and McCann, J.J. (1971). Lightness and retinex theory. *Journal of the Optical Society of America*, 61(1), 1–11. <https://doi.org/10.1364/JOSA.61.000001>.
4. Brlek, S., Labelle, G. and Lacasse, A. (2005). The discrete Green Theorem and some applications in discrete geometry. *Theoretical Computer Science*, 346(2–3), 200–225. <https://doi.org/10.1016/j.tcs.2005.08.019>.
5. Bradski, G. (2000). The OpenCV Library. *Dr. Dobb’s Journal of Software Tools*.