

1 Introduction

This work implements a Line Keep Assist (LKA) perception pipeline using classical computer vision. The system processes driving videos to detect left and right lane boundaries and estimate the vehicle's lateral position. It produces an annotated video and a per-frame CSV log used for quantitative evaluation.

Two datasets were used. The first (`lane.mp4`) shows a straight, well-marked highway. The second (`challenge.mp4`) contains curves, shadows, and faded markings to test robustness under difficult conditions. The implementation follows the assignment baseline: region-of-interest selection, color and gradient thresholding, perspective warp, sliding-window search, polynomial fitting, and temporal smoothing. Each step was coded as a separate module and outputs an intermediate image for inspection.

2 Methodology

2.1 Region of Interest (ROI)

Processing was first restricted to the visible road area using a trapezoidal mask generated in `preprocess.py`. The vertices were defined from the frame size (W, H) as

$$\text{src} = \begin{bmatrix} 0.5W - 0.18W & 0.68H \\ 0.5W + 0.18W & 0.68H \\ W & H \\ 0 & H \end{bmatrix}$$

The top edge corresponded to 68 % of the frame height, excluding the horizon and vehicles ahead. This limited all later operations to the drivable region and avoided false detections above the road. Figure 1 shows the ROI mask where the upper region is fully excluded.

2.2 Pre-processing and Mask Generation

Lane markings were extracted using a combination of color and gradient filters implemented in `threshold_lane_pixels()`. Each frame was converted to grayscale, HLS, and HSV color spaces. From these, the L , S , and V channels were selected for brightness and saturation control. Thresholds were applied as

$$S \in [120, 255], \quad V \in [200, 255], \quad L \in [200, 255].$$

Yellow lanes were detected when $15 \leq H \leq 35$, $S \geq 80$, $V \geq 120$, and white lanes when $S \leq 40$, $V \geq 180$. A Sobel filter in the x direction on L highlighted vertical edges, $G_x = |\frac{\partial L}{\partial x}|$, followed by a threshold in $[30, 255]$. The combined binary mask,

$$M = (M_{\text{yellow}} \vee M_{\text{white}} \vee M_{\text{sobel}}),$$

was multiplied by the ROI to keep only relevant regions. A morphological close-open filter (3×3 kernel) removed small noise, and components smaller than 50 px were discarded. Figures 2–3 show the grayscale frame and the final binary mask used for warping.



Figure 1: Region of interest.



Figure 2: Grayscale frame.



Figure 3: Final binary mask.

2.3 Perspective Warp

Each binary frame was then transformed to a bird–eye view to simplify lane fitting. Four source and destination points were defined as

$$\text{src} = \begin{bmatrix} 0.42W & 0.62H \\ 0.58W & 0.62H \\ 0.90W & H \\ 0.10W & H \end{bmatrix}, \quad \text{dst} = \begin{bmatrix} 0.25W & 0 \\ 0.75W & 0 \\ 0.75W & H \\ 0.25W & H \end{bmatrix}$$

OpenCV’s `getPerspectiveTransform()` returned the matrices (M, M^{-1}) . Applying `cv2.warpPerspective` yielded a top–down view where lane lines appeared parallel. Figures 4 and 5 show the selected region and the resulting warped image.



Figure 4: Selected source region before warping.



Figure 5: Resulting bird–eye view after transformation.

2.4 Lane Pixel Extraction

Lane pixels were extracted from the warped binary image using a sliding–window histogram search in `lane_fit.py`. The lower half of the image was integrated column–wise:

$$H(x) = \sum_{y=H/2}^H I(y, x),$$

and the two histogram peaks marked the initial base positions. Nine search windows of height $h_w = H/9$ and margin $m = 80$ px were then moved upward, recentering when at least 40 active pixels were found. Figure 6 shows the histogram and detected base peaks.

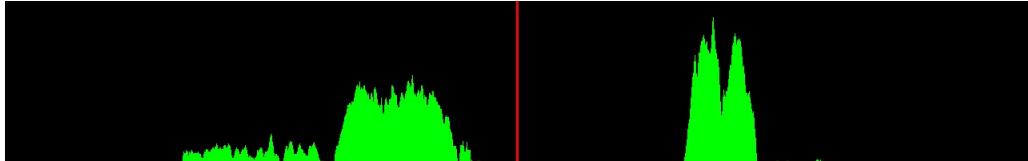


Figure 6: Histogram of the warped binary mask showing lane bases.

2.5 Polynomial Fitting and Confidence

Nonzero pixels collected from both sides were fitted with a second-order polynomial:

$$x(y) = ay^2 + by + c.$$

When pixel coverage dropped below 40 %, a linear fit replaced the quadratic model to maintain stability. Curvature was computed in meters using scaling factors $y_m = 30/720$ and $x_m = 3.7/700$:

$$R = \frac{[1 + (2ay + b)^2]^{3/2}}{|2a|}.$$

Confidence was evaluated as

$$C = 0.6C_{\text{count}} + 0.25C_{\text{res}} + 0.15C_{\text{prior}},$$

and detection was valid when $C > 0.6$. Figure 7 shows the final fitted curves and detected pixels.

2.6 Temporal Smoothing

Frame-to-frame stability was achieved with a temporal filter applied to polynomial coefficients:

$$\hat{p}_t = \alpha p_t + (1 - \alpha)\hat{p}_{t-1}, \quad \alpha = 0.2.$$

If a frame failed to produce a valid detection, the previous fit was reused for up to six frames when its confidence exceeded 0.4. Confidence values decayed exponentially,

$$C_t = \max(C_t, C_{t-1}(1 - \alpha)),$$

which suppressed flicker and preserved continuity across shadows or brief occlusions.

2.7 Overlay and HUD

Detected lane boundaries were projected back into the camera view using M^{-1} . The left and right lanes were drawn in green and blue; dashed gray lines indicated uncertainty. The polygon between them formed the ego-lane region. A head-up display (HUD) overlaid the detection confidence and the lateral offset:

$$\Delta x = \frac{x_c - x_v}{w_p} \times 3.7,$$

where x_c is the lane-center position, x_v the vehicle center, and w_p the lane width in pixels. Figure 8 shows the overlayed output with numerical readouts.

3 Results

3.1 Quantitative Analysis

Each frame was evaluated for lane confidence, lateral offset, and curvature derived from the polynomial coefficients after temporal filtering. Table 1 summarizes detection consistency and lateral stability for both sequences. The metrics show that the straight highway clip maintained near-perfect detection, while the curved and shadowed clip experienced intermittent losses in both confidence and offset stability.

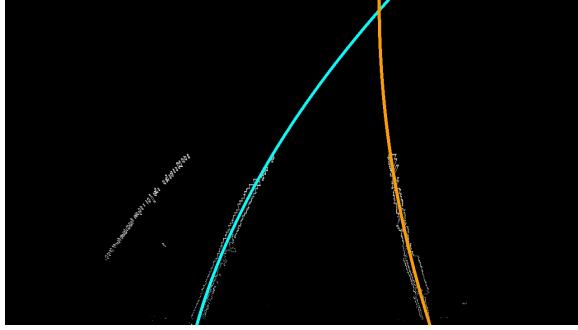


Figure 7: Detected lane pixels and polynomial fits in the warped space.



Figure 8: Final overlay with HUD showing detection confidence and offset.

Table 1: Aggregate metrics computed from per-frame logs. A lane is counted as detected when the per-frame flag equals 1. Mean Conf. is the average of left + right confidences. σ_{offset} is the std. of lateral offset in meters.

Video	Left Det. [%]	Right Det. [%]	Mean Conf.	$\sigma_{\text{offset}} [\text{m}]$
lane.mp4	100.0	100.0	0.94	0.077
challenge.mp4	96.7	99.0	0.91	0.131

Figure 9 shows near-constant confidence around 1.0 for both lanes. The offset remains within $\pm 0.3 \text{ m}$, proving stable center estimation. Curvature spikes appear where edge pixels drop at frame borders but do not affect continuity. The color and Sobel thresholds, together with temporal filtering, ensured smooth tracking throughout.

In Figure 10, confidence fluctuates between 0.4–0.9. Drops around frames 80–150 and 380–450 coincide with shadows and curved sections. Offset error rises to about 0.6 m as the right lane fades, and curvature varies sharply during these periods. Temporal filtering reduced flicker but not full lane loss, showing the weakness of fixed thresholds under poor contrast.

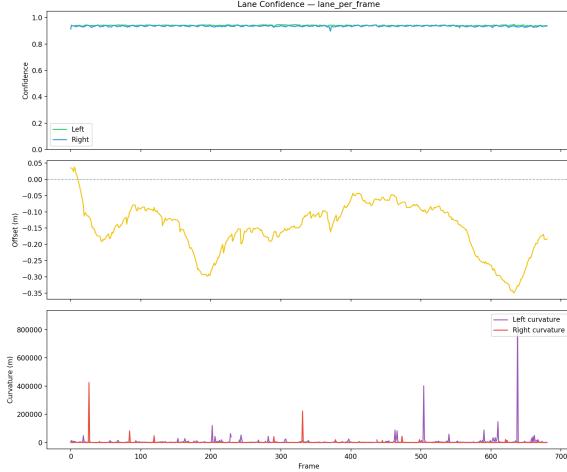


Figure 9: Per-frame confidence, offset, and curvature for `lane.mp4`.



Figure 10: Per-frame confidence, offset, and curvature for `challenge.mp4`.

The highway clip confirms proper parameter tuning for daylight scenes. The curved sequence reveals the limits of planar warping and second-order fitting on strong perspective changes. Performance depends mainly on marking visibility, not algorithmic instability.

3.2 Qualitative Evaluation

Well-Lit Highway. Figure 11 illustrates the system’s behavior on the straight highway clip (`1lane.mp4`). Across frames, both lanes remain stable with confidence above 0.9. The right boundary in Figure 11b shows a minor deviation toward the guardrail where bright asphalt edges entered the ROI, but the temporal filter suppressed any visible drift. In Figure 11c, the pavement alternates between bright and dark sections, yet the HLS and HSV thresholds consistently preserved the valid lane markings while filtering out background noise. This confirms that the chosen ROI and threshold parameters generalize well across illumination changes.



Figure 11: Stable detection and minor edge deviations observed in `1lane.mp4`.

Shadowed Segment. Figure 12 shows three consecutive moments from `challenge.mp4` as the car enters and exits a shadowed region. Before the shadow, both lanes are clearly detected with confidence above 0.9. Inside the dark region, saturation and brightness drop sharply in the S and V channels, breaking the binary mask and causing both confidences to fall to zero. As soon as illumination returns, detection resumes immediately with restored lane curves. This confirms that failures result from lighting variation rather than algorithmic instability.



Figure 12: Shadow transition sequence showing failure and recovery in lane detection.

4 Discussion

4.1 Operational Design Domain (ODD)

The implemented pipeline is valid for daytime highway driving on dry, well-marked asphalt. It assumes a fixed pinhole camera, stable calibration, and clear lane paint inside the defined ROI. Expected illumination range matches the HLS/HSV thresholds used in Section 2.2. Large occlusions, heavy rain, and extreme shadows fall outside this ODD.

4.2 Failure Modes and LKA Action

Loss of either boundary or confidence < 0.6 indicates unreliable perception. In such cases an LKA should disable torque assist and issue a visual and audible warning. If both boundaries are missing for more than $N = 6$ frames (our temporal hold), the system should remain disengaged. Re-engage only when both sides recover with confidence ≥ 0.8 and stable curvature over K frames (e.g., $K = 10$). This prevents steering on spurious edges and matches the drops observed in Figures 10 and 12.

4.3 Insights, Limitations, and Future Work

What worked: ROI, combined color+Sobel mask, and temporal smoothing kept confidence near 0.95 on straight daylight scenes (Figures 9, 11).

Limits: fixed thresholds are contrast-sensitive; planar warp and a single quadratic fit are brittle on strong curves; shadows break the mask (Figure 12).

Next steps: adaptive thresholding (histogram or CLAHE-guided), curvature-aware windows, per-frame exposure normalization, and a light CNN lane mask to replace hand thresholds. Optional fusion with IMU yaw rate and map lane width priors can stabilize curvature and offset during partial loss.

5 Conclusion

This work produced a complete classical vision pipeline for Line Keep Assist perception. The system successfully detected and tracked lane boundaries in daytime highway conditions, maintaining smooth offset and confidence near 1.0. Failures occurred under shadows and curvature (in some cases), where fixed thresholds and planar projection could not preserve lane continuity. Temporal filtering mitigated flicker but did not prevent total loss in low contrast. The results confirm that color-gradient methods remain viable within a narrow ODD but require adaptive thresholding or learning-based segmentation for wider deployment. Future work will test CLAHE-based preprocessing and pretrained lane networks to extend reliability under varied illumination.