Scratch Detection - Approach Document

Document Purpose: To outline the technical strategy, architectural decisions, and roadmap for detecting the scratches on cars and classifying the severity.

Document Metadata

1. **Author:** Deeksha Kashyap
2. **Reviewers:** [Names]
3. **Date:** Monday, 29/12/2025
4. **Version:** v1.0

1. Project Goal

Provide a high-level summary of the release. Define the specific "persona" or domain focus and the intended outcome of the agentic pipeline.

The goal of the project is to detect the scratches on a car and create a classification report based on the severity of the scratches.

2. Current Baseline (State of the Union)

Describe the existing infrastructure to establish what is being built upon.

1. **Technology Stack:** Python, OpenCV, scikit-learn, NumPy, Pandas
2. **Existing Endpoints:** Publicly Available Dataset
3. **Infrastructure:** CPU-based processing

3. Your approach

Provide a technical description of approach and why you chose this. What are the other approaches and how is your approach a better solution for the given problem?

The approach includes a scratch detection and severity classification into two stages

Scratch Detection Module:

- Responsibility: Localize the scratch regions on the car images
- Model: Using a deterministic image processing pipeline
    - Grayscale conversion
    - Gaussian Smoothing
    - Canny edge detection to identify thin, high-contrast structures
    - Morphological dilation
    - Binary thresholding to generate scratch masks

- Output: Binary segmentation masks where white pixels represent detected scratch regions and black pixels represent background

  Severity Label Generation Module:

- Responsibility: Quantify the scratch severity from the binary masks
- Logic: Calculate the scratch area ratio

    Ratio = Scratch pixels/total image pixels

    Discretize into the severity categories:

    - Mild (bottom 33%)
    - Moderate (middle 33%)
    - Severe (top 33%)

Severity Classification Module:

- Responsibility: Classify the images into the categories
- Model: Using a multinomial logistic regression model trained using the scratch area ratio and severity label as the target
- Output: Severity class prediction per image with a standard classification report

4. API Structure & Sample I/O

Document the contract for developers to ensure integration consistency.

Endpoint	**Method**	**Input Description**	**Output/Response Type**
/detect-scratch	POST	Image	Binary mask
/estimate-severity	POST	Binary Mask	Severity label
/evaluate	POST	Dataset	Classification Report

Sample Input/Output Block:

Use this section to provide a raw JSON snippet of a successful request/response cycle.

5. Strategic Roadmap (Phased Approach)

Detail the evolution of the project from the baseline to the target state.

Phase 1: Data Preparation and Preprocessing

1. **Goal:** Ensure image quality and consistency before scratch analysis
2. **Intuition:** Images are sensitive to noise and preprocessing improves edge clarity and reduces false detections
3. **Key Tasks:**
    a. Removal of salt-and-pepper noise using median filtering
    b. Standardization of image resolution and format
    c. Dataset organization into train, validation, and test splits

Phase 2: Scratch Segmentation using Computer Vision

1. **Goal:** Localize scratch-like regions in car images without relying on learned models.
2. **Intuition:** Scratches are thin, high-frequency structures that can be effectively captured through edge-based techniques.
3. **Key Tasks:**
    a. Grayscale conversion
    b. Gaussian smoothing
    c. Canny edge detection
    d. Morphological dilation and thresholding
    e. Generation of binary scratch masks

Phase 3:Severity Quantification & Label Generation

1. **Goal:** Convert segmentation outputs into interpretable severity labels
2. **Intuition:** Pixel-level masks can be reduced to a scalar metric representing damage extent.
3. **Key Tasks:**
    a. Compute scratch area ratio per image
    b. Apply percentile-based binning to derive severity categories:
        i. Low
        ii. Medium
        iii. High
    c. Validate class balance across dataset splits

Phase 4: Severity Classification and Evaluation

4. **Goal:** Evaluate the discriminative power of the severity metric.
5. **Intuition:** A classifier provides a quantitative validation of severity separability.
6. **Key Tasks:**
    a. Train multinomial logistic regression on scratch area ratios
    b. Evaluate on held-out test data

    c. Generate classification report (precision, recall, F1-score)

6. Success Criteria & Metrics

Define how the effectiveness of the new approach will be measured.

1. **Performance:** End-to-end processing latency per image, Successful generation of segmentation masks
2. **Quality:** Classification report
3. **Cost:** Consistent behavior across train, validation and test splits without overfitting.

7. References

1. Canny, J. (1986). A Computational Approach to Edge Detection. IEEE PAMI
2. Gonzalez & Woods. Digital Image Processing. Pearson
3. OpenCV Documentation:
    a. Edge Detection
    b. Morphological Operations
4. scikit-learn Documentation: Classification Metrics