AI R and D Scratch detection- Approach Document

Document Purpose: To outline the technical strategy, architectural decisions, and roadmap for [Core Functionality].

Document Metadata

1. **Author:** Priya Naik
2. **Reviewers:**
3. **Date:** Monday -29-12-2025
4. **Version:** [e.g., v1.0]

1. Project Goal

1. The goal of this project is to build an AI model that detects scratches on surface images using a public dataset.
2. The system classifies images into two categories: Scratch and No Scratch.
3. The solution is designed for quality inspection use cases in manufacturing environments.
4. A CNN-based image classification model is trained and evaluated.
5. The final output includes a classification report and a publicly accessible model and code.

2. Current Baseline (State of the Union)

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

1. **Technology Stack:** Python
2. **Existing Endpoints:** -
3. **Infrastructure:** Built on jupyter environment Validation is performed using standard evaluation metrics and manual inspection of results.

3. Your approach

Agent 1: Scratch Classifier

Classifies images as Scratch or No Scratch using a CNN model.

Agent Name 2: Data Preprocessing Agent

Prepares images by resizing, normalizing, and splitting them into training and testing sets.

Agent 3: Evaluation Agent

Generates accuracy, confusion matrix, and classification report for model validation.

4. API Structure & Sample I/O

Document the contract for developers to ensure integration consistency.

Endpoint	**Method**	**Input Description**	**Output/Response Type**
/scratch-detection/	POST	Image file	JSON with prediction and confidence

Sample Input/Output Block:

INPUT:

file: surface_image.jpg

OUTPUT:

```
{
  "prediction": "Scratch",
  "confidence": 0.92
}
```

5. Strategic Roadmap (Phased Approach)

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

Phase 1: [Optimization/Framework Integration]

1. **Goal:** Improve model performance and structure the code for easy reuse.
2. **Intuition:** clean and optimized pipeline helps in faster experiments and consistent results.
3. **Key Tasks:**

   a.Optimize image preprocessing and training parameters.
   b. Organize code into reusable modules.

4. **Phase 2: [Model Evolution & Annotation]**

1. **Goal:** Extend from classification to scratch localization or segmentation.

2. **Intuition**:Current model only predicts presence of scratch, not its exact location.

3. Key Tasks:

a.Explore pixel-level annotation or bounding box datasets.
b. Train advanced models like U-Net or Mask R-CNN.

Phase 3: [Tooling & Pre-processing]

1. **Goal:** Improve input quality through automated preprocessing tools.

2. **Intuition:** Better image quality leads to better prediction accuracy.

3. **Key Tasks:**

. Add automated cropping, resizing, and noise reduction steps.

6. Success Criteria & Metrics

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

1. **Performance:** Model accuracy, precision, recall, F1-score, and inference time.

2. **Quality:** Stability of predictions across different surface images and datasets.

3. **Cost:** Compute time and resource usage during training and inference.

7. References

1. [Internal/External documentation links]

2. [Benchmark datasets or research papers]