Requirements Gathering – Skin Lesion Detection System

Ngane Emmanuel

2025-07-18

1. Project Overview

The Skin Lesion Detection System is an AI-driven application aimed at detecting and classifying skin conditions from images. It targets both non-technical users (via a mobile application) and technical users such as researchers and developers (via a CLI tool). A support website will provide guides, FAQs, and usage instructions.

---

2. Problem Statement

Early detection of skin conditions, including cancerous lesions, is critical. However, access to dermatologists is limited in many regions, and people often delay consultations due to cost, stigma, or lack of awareness. There's a need for an accessible tool that can assist in identifying potential skin issues from images and recommend follow-up medical action.

---

3. Objectives

- Build an AI model to classify skin lesions as healthy or unhealthy.
- If unhealthy, identify probable diseases and return confidence scores.
- Enable users to receive readable explanations of the results via an LLM.
- Ensure the system is intuitive for general users and powerful for researchers.
- Handle uncertain cases by rejecting inputs outside of training distribution.
- Respect user privacy by avoiding data retention unless explicitly permitted.

---

4. Stakeholders

Stakeholder	Interest
General Users	Want fast, easy-to-use skin health analysis
Researchers	Require reliable, batch-capable CLI output

Stakeholder	Interest
Medical Professionals	Might verify or use insights for triage
Developers/Admins	Maintain system, monitor performance and logs

---

5. User Personas

##**General User (Non-technical)** - Age: 18–60+ - Needs: Simple scan, fast results, privacy - Concerns: Accuracy, trustworthiness, privacy

Researcher

- Role: Data scientist or ML engineer
- Needs: CLI with JSON/CSV output, batch processing
- Concerns: Precision, transparency, reproducibility

Medical Professional

- Role: Dermatologist or general practitioner
- Needs: Condition references, explanation quality
- Concerns: Over-reliance on AI, false negatives

---

6. Functional Requirements

☐ Upload image from mobile app
☐ Classify skin as healthy/unhealthy
☐ Identify and rank probable diseases with confidence scores
☐ Return readable explanation (via LLM)
☐ Reject unknown/out-of-distribution inputs
☐ CLI: Accept JSON/CSV/image folder inputs
☐ CLI: Return raw structured output (JSON/CSV)
☐ Provide Grad-CAM or similar visual explanation (optional)
☐ Admin logging and versioning of model in backend

---

7. Non-Functional Requirements

- Accuracy: >90% for known classes
- Latency: <10s for image inference
- Privacy: Images deleted post-analysis
- Portability: Mobile app (Android/iOS), CLI (Linux/Mac/Windows)
- Explainability: Provide simple user-focused descriptions

- Extensibility: Allow retraining on new datasets

---

8. Use Case Descriptions

Use Case 1: Classify Skin Image (Mobile)

Actor: User
Trigger: Uploads image
Steps: 1. User takes/selects an image of a skin area 2. App uploads to backend 3. Model processes and classifies image 4. Response returned with: - Health status - Possible conditions (w/ confidence) - LLM-generated explanation

Use Case 2: Batch Classification (CLI)

Actor: Researcher
Trigger: Invokes CLI with multiple images or dataset
Steps: 1. CLI tool reads JSON/CSV/image directory 2. Images are processed asynchronously 3. Output is saved locally with raw scores and logs

---

9. Survey Summary *(To be completed after responses)*

This section will include insights derived from: - User feedback (general public) - Researcher/technical requirements - Any expert interviews (if available)

We'll update this section with statistics, charts, or tables summarizing: - Trust levels - Accuracy expectations - Feature demand - Privacy concerns

---