

---

## **Software Requirements Specification (SRS-2)**

**Project Title:** Image Recognition System for Medical Diagnosis (Simulated Data)

**Prepared by:** Sparsh Srivastava

**Date:** 31 October 2025

Link to View:- [Tensorflow](#)

---

### 1. **Introduction**

#### 1.1 Purpose

The purpose of this project is to develop an AI-based image classification system that can analyze simulated medical images (e.g., X-rays, skin lesions) and categorize them into diagnostic classes. This tool supports training and education in medical diagnostics.

#### 1.2 Scope

This system will:

- Accept medical image inputs (simulated datasets).
- Preprocess and classify images using a trained AI model.
- Output diagnostic categories with confidence scores.
- Provide a user interface for uploading and viewing results.

#### 1.3 Definitions, Acronyms, and Abbreviations

- **CNN:** Convolutional Neural Network
  - **AI:** Artificial Intelligence
  - **ML:** Machine Learning
  - **SRS:** Software Requirements Specification
- 

### 2. **Overall Description**

#### 2.1 Product Perspective

The system is a standalone web-based application powered by a backend AI model. It uses TensorFlow/Keras for model training and classification, and Flask or Streamlit for deployment.

## 2.2 Product Functions

- Image upload and preprocessing
- Classification using trained CNN model
- Display of predicted category and confidence
- Option to view sample training data and model performance

## 2.3 User Characteristics

- Medical students, educators, and researchers
- Basic familiarity with web interfaces
- No prior AI knowledge required

## 2.4 Constraints

- Uses simulated data only (no real patient data)
  - Limited to predefined diagnostic categories
  - Requires GPU for training efficiency
- 

## 3. Specific Requirements

### 3.1 Functional Requirements

- **FR1:** The system shall allow users to upload medical images.
- **FR2:** The system shall preprocess images (resize, normalize).
- **FR3:** The system shall classify images into predefined categories.
- **FR4:** The system shall display prediction results with confidence scores.
- **FR5:** The system shall allow viewing model accuracy and confusion matrix.

### 3.2 Non-Functional Requirements

- **NFR1:** The system shall respond within 3 seconds for classification.
- **NFR2:** The model shall achieve at least 85% accuracy on test data.
- **NFR3:** The system shall be accessible via desktop and mobile browsers.

### 3.3 External Interface Requirements

- **User Interface:** Web-based dashboard (Streamlit or Flask)
- **Software Interface:** TensorFlow/Keras, OpenCV, NumPy

- **Hardware Interface:** GPU-enabled training environment (e.g., Google Colab)
- 

4. 🧠 **System Features**

Feature	Description
Image Upload	Users can upload medical images for diagnosis
Preprocessing	Images are resized, normalized, and prepared for model input
Classification	CNN model predicts diagnostic category
Visualization	Results shown with confidence score and optional heatmap
Model Metrics	Accuracy, loss, and confusion matrix displayed for transparency

---

5. 🛠️ **Design Approach**

- **Architecture:** Client-server model with AI backend
  - **Model Type:** CNN (e.g., ResNet, VGG, or custom architecture)
  - **Tools Used:** Python, TensorFlow, Keras, OpenCV, Streamlit/Flask
  - **Dataset:** Simulated medical image dataset (e.g., Kaggle or synthetic)
- 

6. 🚀 **Future Enhancements**

- Integration with real-time camera input
- Expansion to multi-label classification
- Explainable AI features (e.g., Grad-CAM visualizations)
- Integration with electronic health record (EHR) systems