Software Requirements Specification (SRS)

Project: Breast Cancer Detection System
Prepared by: Woroma Dimkpa, Kahlel Cardona, Taratong Dolinsky
Date: 02/17/2026

1. Introduction

1.1 Purpose

The purpose of this software system is to assist medical professionals and researchers in the early detection of breast cancer by providing automated classification of breast tumors as benign (non-cancerous) or malignant (cancerous) using mammogram images from the Curated Breast Imaging Subset of the Digital Database for Screening Mammography dataset. Breast cancer remains one of the most prevalent cancers worldwide and a leading cause of cancer-related deaths among women. Early detection significantly increases the chances of successful treatment and long-term survival. Radiologists and clinicians face challenges such as imaging noise, subtle tumor patterns, and high workloads, which can lead to delayed diagnosis or misclassification of tumors.

The software system aims to improve diagnostic accuracy, reduce the burden on medical professionals, and support informed decision-making in clinical settings. The intended users are medical professionals including radiologists and oncologists, as well as medical researchers who require reliable tools for analyzing mammogram datasets.

1.2 Scope

This system will provide several key functionalities: automatic image preprocessing, CNN-based classification of mammograms, model training and comparison, and visualization of evaluation metrics and misclassified cases. The system will support multiple CNN architectures, including custom-built CNNs and pre-trained models such as ResNet and EfficientNet, utilizing transfer learning. Users can analyze model performance using metrics such as accuracy, precision, recall, and F1-score.

The system is designed to run in Python with support for GPU acceleration, making it suitable for environments such as Kaggle notebooks or local machines equipped with NVIDIA GPUs. The scope includes only mammogram images from the CBIS-DDSM dataset, and the system does not provide direct medical recommendations. This system is intended for research and decision support purposes only and is not approved for direct clinical diagnosis.

1.3 Definitions, Acronyms, and Abbreviations

Term	Definition
CNN	Convolutional Neural Network
GPU	Graphics Processing Unit
CBIS-DDSM	Curated Breast Imaging Subset of Digital Database for Screening Mammography
SRS	Software Requirements Specification
ROI	Region of Interest

1.4 References

1. IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications.
2. CBIS-DDSM dataset, Kaggle: https://www.kaggle.com/datasets/awsaf49/cbis-ddsm-breast-cancer-image-dataset
3. PyTorch Documentation: https://pytorch.org/docs/stable/index.html

2. Overall Description

2.1 Product Perspective

The system is a standalone software solution that follows a typical machine learning pipeline: mammogram images are preprocessed, passed through CNN models for classification, and output is analyzed using evaluation metrics. Visualization tools allow users to understand misclassified cases and model behavior. The system is designed to integrate seamlessly with existing research workflows but is independent of external systems.

2.2 Product Functions

The key functions of the system are as follows:

1. The system shall load mammogram images from the CBIS-DDSM dataset, including metadata and image files.
2. The system shall preprocess images automatically, including resizing, normalization, and optional data augmentation.
3. The system shall train CNN models on either small subsets or the full dataset.

4. The system shall compare multiple CNN architectures, including custom-built and pre-trained models.
5. The system shall apply transfer learning for pretrained models.
6. The system shall classify images as benign or malignant.
7. The system shall compute evaluation metrics, including accuracy, precision, recall, F1-score, and confusion matrices.
8. The system shall visualize model performance and misclassified cases to provide interpretability.
9. The system shall export trained models for later inference.

2.3 User Characteristics

Intended users are expected to have basic computer literacy and familiarity with mammogram images. Medical researchers should understand fundamental machine learning concepts, while radiologists may primarily use the visualization and evaluation functions. The system does not assume users are programmers but provides code-level access for advanced users.

2.4 Constraints

- Availability of GPU resources (e.g. NVIDIA CUDA-enabled GPUs) may be limited in cloud environments such as Kaggle, which may affect model training time.
- Mammogram images are large and may require significant preprocessing to fit into CNN architectures.
- CNN models must be trained efficiently to avoid overfitting due to limited dataset size.

2.5 Assumptions and Dependencies

- CBIS-DDSM dataset is available and correctly formatted in JPEG or DICOM format.
- Python 3.x environment is used with access to libraries including PyTorch, TensorFlow, OpenCV, NumPy, Pandas, and Matplotlib.
- Users have access to GPU resources for efficient model training.

3. Specific Requirements

3.1 Functional Requirements

ID	Requirement	Description
FR1	Image Loading	System shall load mammogram images from the CBIS-DDSM JPEG dataset and associated CSV metadata.
FR2	Preprocessing	System shall automatically resize, normalize, and optionally augment images for model training.
FR3	Model Training	System shall train CNN models on subsets or the full dataset.
FR4	Multi-Architecture Support	System shall allow selection and comparison of multiple CNN architectures.
FR5	Transfer Learning	System shall support transfer learning with pre-trained models.
FR6	Classification	System shall classify each mammogram image as benign or malignant.
FR7	Evaluation Metrics	System shall compute accuracy, precision, recall, F1-score, and confusion matrices.
FR8	Visualization	System shall display sample images, model performance graphs, and misclassified cases.
FR9	Model Export	System shall allow trained models to be saved for future inference.
FR 10	Data Splitting	The system shall split the dataset into training, validation, and test sets with configurable ratios.

3.2 Non-Functional Requirements

ID	Requirement	Description
NFR1	Performance	The system shall provide efficient training and inference performance in GPU-enabled environments. The system shall support batch-based processing and scalable model training on the full CBIS-DDSM dataset without performance degradation or instability.
NFR2	Reliability	The system will handle corrupted, incomplete, or improperly formatted images without crashing. It will log preprocessing errors and model failures for debugging purposes.The system will maintain consistent classification performance across multiple runs when using fixed random seeds.
NFR3	Usability	The system shall generate confusion matrices and training curves (loss and accuracy). The user interface will display classification results in an understandable format (e.g., “Cancerous” or “Non-Cancerous”). Documentation will include setup instructions and example usage workflows.
NFR4	Maintainability	The codebase will be modular, separating preprocessing, model definition, training, and evaluation components. The system will allow easy replacement or addition of CNN architectures without major structural modifications. Source code should include inline comments and documentation.
NFR5	Portability	The system should be compatible with Windows, macOS, and Linux operating systems. The system will run in cloud environments such as Kaggle and Google Colab. Dependencies will be managed using standard package managers such as pip or conda.
NFR6	Reproducibility	The system shall allow configuration of fixed random seeds to ensure reproducible experimental results.

3.3 Interface Requirements

- **User Interface:** Jupyter Notebook interface displaying images, plots, and metrics.
- **Hardware Interface:** GPU acceleration (T4 or P100 recommended).
- **Software Interface:** Python 3.x, PyTorch/TensorFlow, OpenCV, Pandas, Matplotlib.

3.4 Performance Requirements

- Image preprocessing time: ≤ 2 seconds per image.
- Tiny CNN training on subset: ≤ 1 minute.
- Full CNN training: ≤ 1 hour (GPU dependent).

3.5 Design Constraints

- CNN input size: 128×128 for small models, scalable to 224×224 or larger for full models.
- Kaggle environment memory limit: ~16 GB.
- Use only standard Python packages.

3.6 System Attributes

- **Scalability:** System must scale to full CBIS-DDSM dataset and multiple CNN architectures.
- **Robustness:** System must handle missing, corrupted, or unusual images.
- **Reusability:** Modular code structure allows easy adaptation of models or preprocessing techniques.

4. Appendices

A. Dataset Description: CBIS-DDSM dataset includes mammogram images with metadata in CSV format. Each image is labeled as benign or malignant with ROI masks.

B. Sample CNN Architecture: A tiny CNN with two convolutional layers, max pooling, and two fully connected layers is sufficient for initial experiments.

C. Evaluation Metrics: Accuracy, precision, recall, F1-score, confusion matrices, and misclassification visualization.