Milestone 1 Progress Evaluation
Project: Breast Cancer Detection System

Team Members:
Kahlel Cardona - kcardona2023@my.fit.edu
Woroma Dimkpa - wdimkpa2021@my.fit.edu
Taratong Dolinsky - tdolinsky2022@my.fit.edu
Faculty Advisor/Client: Dr. Zahra Nematzadeh - znematzadeh@fit.edu, Department of Computer Science, Florida Institute of Technology
Date: 02/17/2026

Progress Matrix - Milestone 1

Task	Completion %	Kahlel	Woroma	Tara	To Do
1. Compare & Select Technical Tools	100%	Data	Models	Visualization	None
2. "Hello World" Demos	90%	Preprocessing	CNN Training	Metrics	Finalize metrics demo
3. Resolve Technical Challenges	100%	Dataset	Architecture	Evaluation	None
4. Requirements Document (SRS)	100%	50%	25%	25%	None
5. Design Document (SDD)	100%	25%	25%	50%	None
6. Test Plan (STP)	100%	25%	50%	25%	None

Discussion of Accomplished Tasks

Task 1: Compare & Select Technical Tools
The team evaluated and selected tools across three areas, with each member responsible for a specific domain. Kahlel investigated data-handling tools, comparing OpenCV, PIL/Pillow, and torchvision for loading and managing the CBIS-DDSM dataset, and recommended torchvision.transforms for its seamless integration with the PyTorch DataLoader pipeline. Woroma evaluated model training frameworks, comparing PyTorch and TensorFlow/Keras, and recommended PyTorch based on its dynamic computation graph, more transparent debugging workflow, and widespread adoption in academic medical imaging research. Tara evaluated visualization and evaluation libraries, comparing Matplotlib, Seaborn, and scikit-learn, and recommended Matplotlib and Seaborn for training curve and confusion matrix plotting, with scikit-learn for computing accuracy, precision, recall, and F1-score. For collaboration tooling, the

team collectively adopted GitHub for version control, Google Drive for shared documentation, and Discord for communication and task scheduling. No significant obstacles were encountered, as all selected libraries are well-documented and available in the Kaggle and Google Colab environments used for development.

Task 2: "Hello World" Demos

The team developed three proof-of-concept demonstrations to validate the selected toolchain on real CBIS-DDSM data, each corresponding to a team member's assigned domain. Kahlel implemented the preprocessing demo, parsing the CBIS-DDSM CSV metadata, resolving image file paths, loading sample mammogram images, and resizing them to 224x224 pixels, centered on the most dense area of the tumor. Woroma implemented the CNN training demo, constructing a minimal two-layer convolutional network and running a training loop on a small image subset to verify that forward propagation, loss computation, backpropagation, and optimizer updates functioned correctly and produced decreasing training loss. Tara implemented the evaluation metrics demo, applying a trained model checkpoint to a held-out subset and computing accuracy and a basic confusion matrix using scikit-learn. The metrics demo is approximately 90% complete, as the team is finalizing the stratified DataLoader splitting logic needed to ensure class distribution is preserved across train, validation, and test subsets.

Task 3: Resolve Technical Challenges

The team identified and addressed the key technical challenges anticipated in the project plan, each member focusing on their assigned area. Kahlel resolved dataset loading challenges, including mapping CSV metadata fields to image file paths, handling missing or misformatted path entries, and confirming that JPEG images from the CBIS-DDSM dataset could be loaded reliably without runtime errors. Woroma resolved CNN architecture challenges, confirming that a basic convolutional network could be defined and instantiated in PyTorch with correct input and output dimensions for the binary classification task, and that the training loop converged without numerical instability on the small subset. Tara resolved evaluation challenges, verifying that scikit-learn metrics functions produced correct output for both balanced and imbalanced class distributions, and identifying the need for class-weighted loss to address the benign/malignant imbalance present in the CBIS-DDSM dataset. No blocking obstacles remain from this phase; all identified challenges have been documented in the design document and test plan as constraints or risks with mitigation strategies.

Task 4: Requirements Document (SRS)

The Software Requirements Specification (BCD-SRS-1.0) was completed and finalized on February 17, 2026. The document defines ten functional requirements covering image loading, preprocessing, model training, multi-architecture support, transfer learning, classification, evaluation metrics, visualization, model export, and dataset splitting, as well as six non-functional requirements addressing performance, reliability, usability, maintainability, portability, and reproducibility. The document follows the structure recommended by IEEE 830-1998. The primary challenge was precisely scoping the system boundary to clearly establish that the system is a research and decision-support tool and not a clinical diagnostic device.

Task 5: Design Document (SDD)

The Software Design Document (BCD-SDD-1.0) was completed and finalized on February 17, 2026. The document describes a layered modular architecture consisting of six layers: Data, Processing, Model, Training, Evaluation, and Visualization and Persistence. It includes a high-level system architecture diagram, component interaction diagram, training data flow diagram, training sequence diagram, class diagram, and deployment architecture diagram. The primary challenge was determining the appropriate level of design abstraction to guide implementation without over-constraining decisions that may need to evolve during model experimentation.

Task 6: Test Plan (STP)

The Software Test Plan (BCD-STP-1.0) was completed and finalized on February 17, 2026. The plan defines testing across five levels: unit testing, integration testing, system testing, performance testing, and model validation testing. Six functional test cases were specified for the core pipeline and two additional test cases for model export and reload. The plan also defines acceptance criteria including a minimum 75% validation accuracy baseline for the Tiny CNN. The primary challenge was establishing meaningful performance thresholds for GPU-dependent operations given variability in resource availability across Kaggle environments.

Discussion of Team Member Contributions

Kahlel Cardona

Kahlel took ownership of the data domain throughout Milestone 1. For tool selection, Kahlel investigated data-handling and preprocessing libraries, evaluating OpenCV, PIL/Pillow, and torchvision, and recommended torchvision. Transforms for its tight integration with the PyTorch DataLoader. For the Hello World demos, Kahlel implemented the preprocessing demo, confirming that the image loading and transformation pipeline produced correctly shaped, normalized tensors from CBIS-DDSM data. To resolve technical challenges, Kahlel addressed dataset-loading issues, including CSV path mapping. Kahlel contributed to the Requirements Document's functional requirements and scope sections, the Design Document's data loading and preprocessing module descriptions, and the Test Plan's pipeline test cases.

Woroma Dimkpa

Woroma took ownership of the model training domain throughout Milestone 1 and led the drafting of all three documents. For tool selection, Woroma evaluated PyTorch and TensorFlow/Keras and argued for PyTorch based on its dynamic computation graph and research-friendly design. For the Hello World demos, Woroma implemented the CNN training demo, building a minimal two-layer network and verifying that forward propagation, backpropagation, and optimizer updates all functioned correctly on a small subset. To resolve technical challenges, Woroma addressed architectural issues, including confirming the correct input/output tensor dimensions and ensuring training loop stability. Woroma led drafting of the Requirements Document, Design Document, and Test Plan, authoring the majority of content across all three while incorporating input from teammates on their respective domain sections.

Taratong Dolinsky

Tara took ownership of the visualization and evaluation domain throughout Milestone 1. For tool selection, Tara evaluated Matplotlib, Seaborn, and scikit-learn and confirmed their suitability for generating confusion matrices and plotting training curves. For the Hello World demos, Tara implemented the evaluation metrics demo, applying a model checkpoint to a held-out subset and computing accuracy and a confusion matrix using scikit-learn. For resolving technical challenges, Tara identified the effect of class imbalance on evaluation metrics and flagged the need for class-weighted loss in the training engine. Tara contributed to the Requirements Document's non-functional requirements, the Design Document's visualization module description, and the Test Plan's evaluation test cases and schedule.

Plan for Milestone 2 (Mar 30, 2026)

Task	Kahlel	Woroma	Taratong
1. Implement & Test Full Preprocessing Pipeline	Implement the CNN training model	Implement augmentation strategies	Integrate & validate full pipeline
2. Implement & Train Initial CNN Model	Define CNN architecture & training loop	Compute & log evaluation metrics	Visualize training curves & results
3. Implement Transfer Learning	Adapt pretrained model (EfficientNet)	Train & tune transfer learning model	Compare against baseline CNN
4. Evaluate & Compare Initial Models	Run evaluation on test set	Generate confusion matrices & metrics	Analyze & document model comparison
5. Demo Implemented Features	Prepare preprocessing demo	Prepare training & results demo	Prepare evaluation demo

Discussion of Planned Tasks for Milestone 2

Task 1: Implement and Test Full Preprocessing Pipeline

The full preprocessing pipeline will be implemented and validated across the complete CBIS-DDSM dataset, extending the proof-of-concept from Milestone 1 into a production-ready component. This includes robust error handling for corrupted or missing image files, stratified train/validation/test splitting with configurable ratios, and data augmentation strategies such as horizontal flipping and brightness adjustments to improve model generalization. The pipeline must satisfy the preprocessing performance threshold of 2 seconds or less per image as defined in the SRS. Each component will be unit tested and integration tested before the full pipeline is validated end-to-end from raw CSV input to batched tensor output.

Task 2: Implement and Train Initial CNN Model

A baseline CNN model will be implemented and trained on the full dataset. This extends the minimal Hello World demo into a complete training loop with configurable hyperparameters, class-weighted loss to address the benign/malignant class imbalance, and validation monitoring to detect overfitting. Model checkpointing will be implemented to save the best-performing weights during training. Training and validation loss and accuracy will be logged per epoch and used to assess convergence. This model will serve as the baseline against which transfer learning architectures are compared in Task 3.

Task 3: Implement Transfer Learning

At least one pre-trained architecture, either ResNet or EfficientNet, will be adapted for the breast cancer classification task. This involves modifying the first convolutional layer to accept single-channel mammogram input and replacing the final classification head with a two-class output layer. Transfer learning weights will be fine-tuned on the CBIS-DDSM training set. The resulting model will be evaluated using the same metrics and test split as the baseline CNN to enable a direct comparison. This task directly implements SRS functional requirements FR4 and FR5.

Task 4: Evaluate and Compare Initial Models

Both the baseline CNN and the transfer learning model will be evaluated on the held-out test set using accuracy, precision, recall, F1-score, and confusion matrices. Results will be compared side by side to identify which architecture performs better and to understand where each model succeeds or struggles. Misclassified images will be reviewed to look for patterns or dataset-level challenges. The findings will be documented to inform the refinement and analysis work planned for Milestone 3.

Task 5: Demo Implemented Features and Intermediate Results

The team will prepare and present a demonstration of all implemented features at the Milestone 2 checkpoint. This includes a walkthrough of the preprocessing pipeline, a demonstration of training and validation curves for both the baseline CNN and transfer learning model, and a presentation of evaluation metrics and confusion matrices. The demo will highlight intermediate results and any observations about model behavior or dataset challenges encountered during implementation. Feedback from the faculty advisor will be incorporated into the Milestone 3 refinement plan.

Meetings and Faculty Advisor Feedback

Dates of Meetings with Client/Faculty Advisor

January 15, 2026

February 12, 2026

Faculty Advisor Feedback on Milestone 1 Tasks

Task 1 (Compare & Select Technical Tools): ...

Task 2 ("Hello World" Demos): ...

Task 3 (Resolve Technical Challenges): ...

Task 4 (Requirements Document): ...

Task 5 (Design Document): ...

Task 6 (Test Plan): ...

Faculty Advisor Signature: _____ Date: _2/23/26_____

Evaluation by Faculty Advisor

Faculty Advisor: detach and return this page to Dr. Chan (HC 209) or email the scores to pkc@cs.fit.edu

Score (0–10) for each member: circle a score (or circle two adjacent scores for .25, or write a real number between 0 and 10)

Team Member	**0**	**1**	**2**	**3**	**4**	**5**	**5.5**	**6**	**6.5**	**7**	**7.5**	**8**	**8.5**	**9**	**9.5**	**10**
Kahlel Cardona																
Woroma Dimkpa																
Taratong Dolinsky																

Faculty Advisor Signature: _____ Date: __2/23/26_____