Milestone 2 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: 03/30/2026

Progress Matrix - Milestone 2

Task	Completion %	Kahlel	Woroma	Tara	To Do
1. Implement & Test Full Preprocessing Pipeline	100%	CNN training integration	Augmentation strategies	Pipeline validation & testing	None
2. Implement & Train Initial CNN Model	100%	Define architecture & training loop	Compute & log evaluation metrics	Visualize training curves & results	None
3. Implement Transfer Learning	70%	Adapt pretrained models	Begin training transfer learning models	Initial comparison against baseline	Complete training runs; full evaluation s
4. Evaluate & Compare Initial Models	40%	Run partial evaluation on validation set	Generate preliminary metrics	Begin comparison documentation	Full side-by-side comparison
5. Demo Implemented Features	100%	Preprocessing demo	Training & results demo	Evaluation demo	None

Discussion of Accomplished Tasks

Task 1: Implement and Test Full Preprocessing Pipeline

The full preprocessing pipeline was implemented and validated across the CBIS-DDSM dataset, extending the proof-of-concept developed in Milestone 1 into a stable and reusable training pipeline. The preprocessing system now supports automated loading of mammogram metadata

from CSV files, image-path matching, tensor conversion, and normalization for downstream model training.

Kahlel Cardona finalized integration of the preprocessing pipeline with the transfer learning training workflow, ensuring that correctly shaped and normalized tensors are consistently generated for the EfficientNet model through the DataLoader. This included configuring image resizing, normalization, and augmentation for pretrained model compatibility.

Woroma implemented preprocessing support for the Tiny CNN pipeline, adapting image preparation to match the simpler custom convolutional architecture and verifying that data could be passed correctly into the baseline model.

Tara validated the end-to-end preprocessing pipeline for both architectures and confirmed that batched outputs remained consistent across train, validation, and test splits. Tara also verified that class distributions were preserved during splitting to reduce sampling bias during evaluation.

Task 2: Implement and Train Initial CNN Model

Two separate convolutional models were implemented and trained on the full CBIS-DDSM dataset: a baseline Tiny CNN and a transfer learning model based on EfficientNet.

Woroma developed and trained the Tiny CNN architecture, including implementation of the baseline training loop, configurable hyperparameters, and early model evaluation. To address class imbalance between benign and malignant cases, class-weighted loss was incorporated into Tiny CNN training, and per-epoch loss and accuracy were logged to monitor convergence.

Kahlel Cardona implemented the EfficientNet-B0 transfer learning model using pretrained weights. The classifier head was replaced with a two-class output layer for benign and malignant prediction, and all backbone layers were initially frozen so only the classifier trained during early epochs. To improve generalization, 5-fold cross-validation was added using scikit-learn, ensuring each fold preserved benign/malignant class balance. A weighted random sampler was introduced during training so malignant samples were drawn more frequently within each batch, reducing the effect of class imbalance.

Tara produced training visualizations for both models using Matplotlib, including loss curves and validation accuracy plots, allowing direct comparison of convergence behavior between the Tiny CNN and EfficientNet approaches.

The Tiny CNN achieved validation performance near the minimum threshold defined in the Software Test Plan, while EfficientNet demonstrated stronger early validation performance due to transfer learning and pretrained feature extraction.

Task 3: Implement Transfer Learning (In Progress)

The transfer learning pipeline was upgraded beyond basic classifier replacement to include staged fine-tuning of pretrained layers.

Kahlel Cardona implemented gradual unfreezing of EfficientNet backbone layers during training. The final MBConv block is unfrozen after several epochs, followed later by the second-to-last block, allowing pretrained features to adapt progressively without destabilizing early learning. Differential learning rates were introduced through AdamW optimization. The classifier uses a higher learning rate, while newly unfrozen pretrained layers use a lower rate to preserve previously learned features. Weight decay was also added to reduce overfitting. A scheduler was implemented to lower learning rates automatically when validation loss stops improving, improving stability during longer training runs.

Woroma continued refinement of Tiny CNN performance so that transfer learning improvements could be compared against a stable baseline.

Tara monitored fold-by-fold learning curves to identify overfitting and compare convergence speed between both architectures.

Full evaluation and comparison against the baseline will be carried into Milestone 3.

Task 4: Evaluate and Compare Initial Models (In Progress)

Evaluation was standardized so both models could be measured under the same metrics.

Kahlel Cardona evaluated EfficientNet using fold-level validation loss, accuracy, precision, recall, and F1-score during each epoch of cross-validation. The best-performing model from each fold is saved automatically as a checkpoint.

Woroma evaluated Tiny CNN using equivalent classification metrics to provide a baseline comparison.

Tara generated learning-curve plots for every fold, including separate saved graphs for training and validation loss as well as training and validation accuracy.

Cross-validation now produces both mean accuracy and standard deviation across folds, giving a stronger measure of model consistency than a single validation split.

Task 5: Demo of Implemented Features

The team prepared and presented a demonstration of all completed components. Kahlel Cardona demonstrated the EfficientNet training pipeline, including raw dataset construction, fold splitting, weighted sampling, gradual unfreezing, optimizer scheduling, and checkpoint saving. Woroma presented the tiny CNN training run, showing loss and accuracy curves across epochs and discussing how class-weighted loss affected training stability. Tara presented the confusion matrix and evaluation metrics for the baseline model and showed the current state of the transfer learning training runs. The demo made clear what has been completed and what work remains through the rest of the project.

Discussion of Team Member Contributions

Kahlel Cardona

Kahlel focused on the transfer learning pipeline and preprocessing integration. This included building the raw dataset loader, integrating preprocessing into the 5-fold cross-validation workflow, and implementing the EfficientNet model for benign/malignant classification. Kahlel also added checkpoint saving, weighted sampling support, and gradual unfreezing of pretrained layers during training to improve model fine-tuning and generalization

Woroma Dimkpa

Woroma focused on the Tiny CNN baseline model, maintaining and refining its training pipeline so it could serve as the project's comparison architecture. This included ensuring the baseline model remained compatible with the shared preprocessing framework and supporting class imbalance handling during training.

Taratong Dolinsky

Tara continued ownership of evaluation and visualization. Tara validated the full preprocessing pipeline end-to-end, implemented training curve visualizations, produced the baseline confusion matrix and evaluation metrics, and set up the model comparison framework that will be fully populated in Milestone 3. Tara also documented initial observations about baseline model misclassifications to guide analysis in upcoming milestones and led the evaluation portion of the demo.

Plan for Milestone 3 (Apr 20, 2026)

Task	Kahlel	Woroma	Taratong
1. Complete Transfer Learning Training & Initial Evaluation	Finalize EfficientNet training runs and add ResNet	Tune learning rate and batch size	Run evaluation on test set for both models
2. Compare model performance	Analyze fold-level EfficientNet accuracy, precision, recall, and F1-score	Summarize Tiny CNN performance under identical evaluation metrics	Produce side-by-side performance visualizations for all models
3. Improve Transfer Learning Performance	Tune learning rate, dropout, and layer unfreezing schedule	Tune learning rate, dropout, and layer unfreezing schedule	Track changes through updated learning-curve plots
4. Update Documentation	Update SDD to reflect implemented architecture	Document training results and observations	Update test plan with new test cases

Discussion of Planned Tasks for Milestone 3

Task 1: Complete Transfer Learning Training and Initial Evaluation

Task 1 will complete the remaining 5-fold cross-validation runs and finalize model checkpoints so that both architectures have stable evaluation results. This will establish the final validation metrics used for comparison. Also, adding an additional CNN to compare against the others.

Task 2: Compare Model Performance

Task 2 will focus on comparing both models using accuracy, precision, recall, F1-score, and validation loss. This will determine whether transfer learning provides a measurable improvement over the baseline CNN.

Task 3: Improve Transfer learning Performance

Task 3 will continue improving transfer learning performance through hyperparameter tuning, including learning rate adjustments, dropout changes, and refinement of the layer unfreezing schedule. The goal is to improve generalization while reducing overfitting.

Task 4: Update Documentation

The SDD and test plan will be updated to reflect decisions made during implementation. Training results and any deviations from the original design will be documented to maintain traceability to the SRS.

Meetings and Faculty Advisor Feedback

Dates of Meetings with Client/Faculty Advisor

January 15, 2026

February 12, 2026

March 2, 2026

March 16, 2026

Faculty Advisor Feedback on Milestone 2 Tasks Task

1 (Full Preprocessing Pipeline): ...

Task 2 (Initial CNN Model): ...

Task 3 (Transfer Learning): ...

Task 4 (Model Evaluation & Comparison): ...

Task 5 (Demo): ...

Faculty Advisor Signature: _____ Date: _____

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: _____