

BreastCancerDetectionCNN

Milestone 5 Progress Evaluation — Fall 2026

Team

Role	Name	Email
Member	Kahlel Cardona	kcardona2023@my.fit.edu
Member	Woroma Dimkpa	wdimkpa2021@my.fit.edu
Member	Taratong Dolinsky	tdolinsky2022@my.fit.edu
Faculty Advisor	Dr. Zahra Nematzadeh	znematzadeh@fit.edu

Summary

Milestone 5 continued the accuracy work that Milestone 4 showed was paying off. Eight full 5-fold training runs and ten short lesion-patch pretraining runs were completed (about 94 GPU-hours). Every model was chosen on validation data and then measured once on the untouched 704-image held-out test set (428 benign, 276 malignant).

- **Best results of the project, for both models.** ResNet-50 (5-fold ensemble): held-out accuracy 0.774, precision 0.693, recall 0.761, F1 0.725, against the semester baseline of 0.678 / F1 0.6534. EfficientNet-B0 (5-fold ensemble): accuracy 0.751, F1 0.703, recall 0.750, against its baseline of 0.671 / F1 0.6375. Both F1 gains (+0.07) are well beyond the roughly 0.02–0.03 noise of a 704-image test set.
- **Higher resolution helped up to 1536×864, then stopped.** Training at 1024×576 raised held-out F1 by +0.03 (ResNet) and +0.02 (EfficientNet) over 512×288. 1536×864 raised validation F1 further. 2048×1152 was worse for both models, so 1536×864 is the chosen size.
- **Lesion-patch pretraining was the largest single gain.** Each backbone is first trained to tell normal tissue, benign lesions and malignant lesions apart on small patches cut around the dataset's marked lesions, then trained on whole mammograms. At the same 1536×864 size this raised held-out F1 by +0.037 (ResNet) and +0.035 (EfficientNet).
- **Web app brought up to date.** The app now shows a Grad-CAM heatmap next to each verdict, has drag-and-drop upload, a loading indicator and mobile layout, and serves the patch-pretrained 1536×864 models in 16-bit. It returns a result with both heatmaps in about 1.1 seconds. Still open: the 10-trial reliability batch and the poster.

Progress of Current Milestone (Progress Matrix)

Completion percentages are based on what is in the repository as of October 26. If any of this work was done outside the repository (for example, the poster), update the row. Per-member splits are left for the team to fill in (highlighted).

Task	Completion %	Kahlel	Woroma	Tara	To do
1. Integrate the heatmap overlay into the web app; decide on switching the app's EfficientNet	100%	15%	85%	0%	None. gradcam.py heatmaps shown in the app; app switched to the patch-pretrained 1536×864 models
2. Full evaluation suite: held-out accuracy/F1, upload-to-result latency, reliability trial batch	75%	75%	0%	0%	Held-out accuracy/F1 done for every run, and re-checked through the app's code. Latency measured (1.1 s). The 10-trial reliability batch remains
3. Senior Design Showcase poster	0%	0%	0%	0%	Poster preparation pending. Will include project overview, objectives, methodology, system architecture, model evaluation results, Grad-CAM heatmap examples, and final conclusions.
4. Final Answer respects each model's threshold; try a higher training resolution	100%	100%	0%	0%	None. Both served models now use a 0.5 threshold, so the Final Answer can no longer contradict two agreeing verdicts
5. Carry-over: Milestone 4 Tasks 2/3 (overlay, drag-and-drop, loading indicator, mobile CSS, CompareModels.py)	100%	0%	60%	40%	None
Added: 16-bit mixed precision, gradient accumulation, loader-worker option (both training scripts)	100%	100%	0%	0%	None
Added: resolution ladder 1536×864 and 2048×1152, both models	100%	100%	0%	0%	None
Added: lesion-patch pretraining (make_patches.py, PatchPretrain.py, --patch-init) and patch-initialised runs	100%	100%	0%	0%	

Added: user accounts, login security (hashed passwords, CSRF, sessions, logout) and private reading history (SQLAlchemy, SQLite)	100%	0%	100%	0%	None
Added: upload safeguards: mammogram-validity check, metadata removal, thumbnail-only storage, terms of use and authorization checkbox	100%	0%	100%	0%	None
Added: single-answer result view with optional Technical view (advisor feedback)	100%	0%	100%	0%	None
Added: web app security hardening and 18 automated tests	100%	0%	100%	0%	None

Discussion of Each Accomplished Task

Task 4 (part): Higher training resolution, 1024×576

Milestone 4 found that image loading, not the GPU, set the training time, so a larger image size was expected to be cheap. Memory and speed probes picked 1024×576 with a batch of 8 for both models. It keeps the mammograms' shape and is twice as fine along each side as 512×288 (about 0.25 mm of tissue per pixel instead of 0.5 mm), which matters for microcalcifications, which are only 0.1–0.5 mm across.

Both runs finished. Mean validation F1 rose from 0.711 to 0.738 (ResNet) and from 0.679 to 0.740 (EfficientNet), and the held-out set agreed: the ResNet ensemble reached accuracy 0.744 and F1 0.690 (from 0.713 / 0.660), and the EfficientNet ensemble 0.717 / 0.667 (from 0.709 / 0.647). The EfficientNet run stalled in its last fold when Windows ran out of paging-file memory. The four finished folds and fold 5's best checkpoint so far were kept, and fold 5 was scored separately on its exact validation split (see Obstacles).

Added task: 16-bit mixed precision and gradient accumulation

Above 1024×576, the normal 32-bit training no longer fits in the memory available on the 8 GB card. Both training scripts gained three opt-in options: `--amp` (run most arithmetic in the 16-bit bfloat16 format, roughly halving activation memory), `--accum-steps` (split each batch of 8 into smaller pieces and add up their gradients before one update) and `--num-workers`. A CPU check confirmed that one batch of 8 and two accumulated batches of 4 give the same gradients to within 1e-15, including both steps of the SAM optimiser.

While the 1536×864 results were being checked, a 16-bit-trained ResNet lost 0.05 validation F1 when re-scored in 32-bit. A threshold-free check (ROC-AUC) showed this is a real effect, not rounding: every model ranks images best in the precision it was trained in (for example, ResNet 1536×864 AUC 0.891 in 16-bit vs. 0.855 in 32-bit; the 32-bit-trained 1024×576 ResNet 0.875 in 32-bit vs. 0.859 in 16-bit). From then on, every run is scored in its training precision, `evaluate.py` gained an `--amp` option, and the web app must run 16-bit models in 16-bit.

Added task: Resolution ladder, 1536×864 and 2048×1152

The team chose to keep raising resolution one step at a time, separately for each model, until a step stopped improving mean validation F1 by at least 0.005 or the hardware could not run it. Test-set results were recorded but did not decide anything.

- **1536×864 (16-bit, batch 8):** validation F1 improved for both (ResNet 0.738 → 0.767, EfficientNet 0.740 → 0.761). On the held-out set it tied 1024×576 for both models (ResNet ensemble F1 0.688, EfficientNet 0.668), within noise.
- **2048×1152 (16-bit, batch 4 with 2 accumulation steps, forced by memory):** worse for both on validation (ResNet 0.736, EfficientNet 0.721) and on held-out data (ensemble F1 0.653 and 0.627). Both ladders stopped and 1536×864 became the chosen size. The smaller batch at this size may have contributed; separating the two effects would need a run that does not fit the GPU.

Added task: Lesion-patch pretraining

A lesion covers well under 1% of a mammogram, so a network that only sees whole images with one label each gets a weak signal about where to look. Following the approach published for this dataset (Shen et al., 2019), `make_patches.py` uses the dataset's lesion masks to cut 288×288 patches around every marked lesion, plus up to three normal-tissue patches per mammogram. All patches are cut from images resized to 1536×864, so lesions look the same size as in whole-image training. This produced 7,096 normal, 1,633 benign and 1,165 malignant patches in about 10 minutes (66 lesion rows, 2.3%, could not be cut). A by-eye check of 20 patches per class passed.

`PatchPretrain.py` trains each backbone on these three classes once per cross-validation fold, using only that fold's training patients so no validation lesion is seen early (about an hour per model). The `--patch-init` option then starts whole-image training from those weights, with every other setting identical to the plain 1536×864 run. EfficientNet's patch stage missed one of its acceptance checks in three folds (accuracy margin over always guessing "normal"), while lesion F1 passed in every fold; the team decided to run it anyway, and the whole-image result confirmed the benefit.

Patch pretraining raised validation F1 for both models (ResNet 0.767 → 0.777, EfficientNet 0.761 → 0.777, higher in every EfficientNet fold) and gave the largest held-out gain of the semester: ResNet ensemble F1 0.688 → 0.725 and accuracy 0.740 → 0.774; EfficientNet ensemble F1 0.668 → 0.703 and accuracy 0.723 → 0.751. ResNet's folds vary more after pretraining (validation F1 0.70 to 0.82).

Tasks 1 and 5: Heatmap overlay and web app polish

gradcam.py computes a Grad-CAM heatmap for the class each model voted for, from ResNet's last convolutional block and EfficientNet's last feature block. For a 5-fold ensemble it averages the five maps. The heatmap is drawn over the original mammogram, at the original aspect ratio, and shown next to that model's verdict. If a heatmap fails, the verdicts are still returned. The page was redesigned (templates/index.html, static/style.css). static/upload.js adds drag-and-drop, an image preview and a loading state, and the stylesheet has phone and wide-screen layouts. CompareModels.py no longer refers to the missing plots_distill_kfold folder. validate.py (a check that rejects colour photos, blank images and odd shapes) was added but is not yet called by the app.

Tasks 1 and 4: Serving the patch-pretrained models

eval_results/selected_models.json now points at the two patch-pretrained 1536×864 5-fold ensembles, each with a 0.5 threshold. A new optional "amp" field makes inference.py and gradcam.py run a model in 16-bit, the precision it was trained in. Models without the field run exactly as before, which was confirmed by re-running the old models on four test images. Run through the app's own code, the new models reproduced the recorded held-out results exactly on all 704 images (ResNet accuracy 0.774, F1 0.725; EfficientNet 0.751, F1 0.703). An upload with both heatmaps takes a median of 1.10 seconds (peak GPU memory 1.8 GB). The no-file, text-file and fake-image uploads still show the right error messages. The heatmaps are now focused: on a malignant test image, both models mark the same small spot inside the breast, while the old models also marked the nipple and the black background.

Because the old EfficientNet threshold (0.302) is gone, the page can no longer show "Malignant" with a confidence below 50%, and the Final Answer (an accuracy-weighted average of the two models) can no longer contradict two agreeing verdicts.

Task 2 (part): Held-out evaluation of every run

All numbers are on the same 704 held-out images, with test-time augmentation (flips) and a 0.5 threshold, scored in each run's training precision. "Single best" is the fold with the highest validation F1, chosen without looking at the test set. Rows marked (M4) are from Milestone 4.

Configuration	Mean val F1	Single best acc / F1	Ensemble acc / prec / rec / F1	Train time
ResNet-50, 512×288 (M4, previously served)	0.711	0.715 / 0.661	0.713 / 0.616 / 0.710 / 0.660	7h 17m

ResNet-50, 1024×576	0.738	0.733 / 0.690	0.744 / 0.658 / 0.725 / 0.690	≈18h
ResNet-50, 1536×864, 16-bit	0.767	0.714 / 0.656	0.740 / 0.650 / 0.732 / 0.688	12h 54m
ResNet-50, 2048×1152, 16-bit	0.736	0.707 / 0.680	0.697 / 0.593 / 0.725 / 0.653	15h 15m
ResNet-50, 1536×864 + patch pretraining (now served)	0.777	0.781 / 0.728	0.774 / 0.693 / 0.761 / 0.725	12h 35m + 1h
EfficientNet-B0, 512×288 (M4)	0.679	0.713 / 0.638	0.709 / 0.616 / 0.681 / 0.647	6h 18m
EfficientNet-B0, 1024×576	0.740	0.716 / 0.667	0.717 / 0.620 / 0.721 / 0.667	≈7h 15m*
EfficientNet-B0, 1536×864, 16-bit	0.761	0.739 / 0.670	0.723 / 0.630 / 0.710 / 0.668	7h 43m
EfficientNet-B0, 2048×1152, 16-bit	0.721	0.723 / 0.656	0.696 / 0.604 / 0.652 / 0.627	10h 20m
EfficientNet-B0, 1536×864 + patch pretraining (now served)	0.777	0.754 / 0.704	0.751 / 0.661 / 0.750 / 0.703	8h 22m + 1h

** Fold 5 stalled at epoch 33 of 35 and was salvaged; see Obstacles. Patch pretraining time is for all five folds of one model.*

Added Task: User accounts, login security and reading history

The intended users are radiologists, clinicians, researchers and students, not one-time visitors, so the app gained accounts and a private history of past readings. It uses Flask with SQLAlchemy and an SQLite database; because the code goes through SQLAlchemy, moving to PostgreSQL for a deployment only means changing the database setting. Registration records the user's role and the version of the terms of use accepted.

Login security: passwords are stored only as salted hashes; the session secret key must be supplied as an environment variable (the app refuses to start without one outside debug mode); every form carries a CSRF token; cookies are HttpOnly and SameSite, and Secure outside debug mode; sessions expire after 30 minutes; five failed logins lock an account for 15 minutes; the login error message is the same whether or not the email exists; and redirects after login are limited to pages on the same site. Database access uses SQLAlchemy queries only. A user can open or delete all of their own readings.

Added Task: Upload safeguards and privacy

The earlier app returned a score and heatmap for any image, and uploads cannot be guaranteed to be de-identified. Three safeguards were added.

- **Mammogram check.** Uploads are rejected, with the reason shown, if they are in colour, blank, have too few gray levels, are smaller than 128 pixels on a side or have an unusual shape. Little dark background only produces a note, because the models were also trained on cropped regions. These are heuristics that filter out obviously wrong images; they do not prove an image is a mammogram.
- **Metadata removal.** Each upload is decoded and rebuilt from its pixels alone, which drops EXIF, ICC profile, PNG text and Photoshop data, and the user is told what was discarded (a test image showed an ICC profile and Photoshop data). DICOM files are rejected for now, and very large images are refused to guard against decompression bombs. Text burned into the image pixels cannot be removed automatically.
- **Minimal storage.** The original upload and its file name are never saved. Only the result, the confidence values and a small re-encoded thumbnail (320 pixels at most) are stored, and the user can delete any reading or all of them. A Terms of Use page states that only authorized data should be uploaded, and each upload requires a confirmation box.

Added Task: Single answer result view and technical view

The faculty advisor advised against showing both models' answers to non-technical users. By default the result page now shows one answer ("Likely benign" or "Suspicious for malignancy") with a confidence value and a plain caution that the tool is a research prototype and not a diagnosis. If the two models disagree, the page says the analysis was uncertain without exposing the two separate answers. A Technical view switch in the sidebar adds a model choice (blended answer, ResNet-50 only or EfficientNet-B0 only) and a details panel with each model's malignancy probability, whether they disagree, and whether metadata was discarded. The pages were restyled with the colors and login layout from Tara's physician portal design.

Added Task: Web app security hardening and automated tests

The app sends security headers, prevents caching of signed-in pages, caps uploads at 15 MB, and no longer runs in debug mode or listens on all network interfaces by default. 18 automated tests (pytest) cover login and CSRF protection, password hashing, account lockout, redirect safety, SQL-injection input, rejection of colour, blank and non-image uploads, metadata removal and privacy of stored data, separation of one user's history from another's, and the single-answer versus Technical view. They run against a stand-in predictor, so they test the web app and not the models; the models were checked by hand with a real upload. Not yet tested: concurrent users, response time with the new models, and a wider set of images.

Evaluation against the semester success criteria

- **Accuracy / F1: met and exceeded for both models.** ResNet-50 F1 0.725 vs. baseline 0.6534 (+0.072), accuracy 0.774 vs. 0.678. EfficientNet-B0 F1 0.703 vs. 0.6375 (+0.065), accuracy 0.751 vs. 0.671. Recall also rose (0.761 and 0.750), so the gains do not come from

missing more cancers. The validation-vs-test gap (Challenge 2) is now 0.05–0.07 F1, against validation F1 0.70–0.75 and test F1 0.55–0.65 in the project plan.

- **Reproducibility: met.** Seven of the eight runs ended with empty error logs. All runs have five checkpoints of the expected size (107,226,339 bytes ResNet, 16,342,523 bytes EfficientNet). The only exception, the EfficientNet 1024×576 stall, is documented and was handled by a recorded rule.
- **Speed: met.** A full upload, with both 5-fold ensembles at 1536×864 in 16-bit and both heatmaps, takes a median of 1.10 seconds (range 1.1–1.6 s), under the 2-second target. Milestone 4 measured 0.25 seconds at 512×288 without heatmaps.
- **Reliability: partly checked.** The no-file, text-file and fake-image uploads give the right error messages on the new models, and the server log shows no heatmap errors. The full 10-trial batch moves to Milestone 6.

Obstacles

- Windows memory limits. The EfficientNet 1024×576 run stalled when a data-loader worker could not start ("The paging file is too small"); each worker holds about 1.7 GB of memory. The run was salvaged rather than repeated, and later EfficientNet runs use 4 loader workers instead of 8.
- The 16-bit precision effect cost about a day of re-scoring and a decision on how to compare runs fairly. One re-scoring script also hung for almost 2 hours because Windows restarts the script in every loader worker unless it has a main guard.
- Runs took longer than projected (ResNet 1536×864: 12h 54m against 9.6h) because image decoding on the CPU limits speed; GPU use was only about 43%. The eight runs filled most of the milestone, one at a time on the single GPU.
- The Final Answer now scores below ResNet on its own (held-out accuracy 0.757 and F1 0.713, against ResNet's 0.774 and 0.725). The blend weighs the two models by their accuracy, so the slightly weaker EfficientNet pulls the combined answer down. How to combine them is a Milestone 6 decision.

Discussion of Each Team Member's Contribution

Kahlel Cardona: Implemented lesion pre-training and tested how increasing resolution affected f1 score.

Woroma Dimkpa: Integrated, tested, and debugged the web app. Connected Kahlel's selected 5-fold ensembles through inference.py, got the Grad-CAM heatmap working end to end on real uploads, added user accounts with login security, a private reading history that stores only results and thumbnails, a mammogram-validity check and metadata removal on upload, terms of use with a per-upload authorization checkbox, and a single-answer result view with an optional Technical view.

Taratong Dolinsky: Contributed to the design and development of the web application's user interface. Helped create and organize the webpage layout, login page, and dashboard to improve usability and navigation. Also worked on developing a chat box feature to allow users to interact with an AI assistant. Focused on making the application more user-friendly, visually appealing, and accessible.

Plan for Next Milestone (Milestone 6, Nov 23) — Task Matrix

Task	Kahlel	Woroma	Tara
1. Decide how the Final Answer combines the two models (it currently scores below ResNet alone), and call validate.py's image check from the upload path	80%	20%	0%
2. Add Chatbox feature to allow users interact with AI	0%	0%	100%
3. Test and demo the whole system end to end (web app and CompareModels.py) on a fresh machine	0%	0%	100%
4. Final evaluation: semester before/after for both models, latency and the 10-trial reliability batch on the served models	80%	20%	0%
5. User / developer manual (expand README)	0%	0%	100%
6. Demo video of the upload-to-verdict flow, with heatmaps	0%	100%	0%

Discussion of Each Planned Task for Milestone 6

Task 1: Compare a few ways of combining the two models on validation data (for example ResNet alone, equal weights, or weights chosen on validation F1) and keep the one that scores best, then confirm it once on the held-out set. Have the app reject non-mammogram uploads with validate.py, with a clear message.

Task 2: Install the project from the repository on a second machine, following only the README, and confirm the app and dashboard run. Every setup gap found becomes a manual fix.

Task 3: Produce the full semester table (baseline through Milestone 6) for both models, measure upload-to-result time against the 2-second target at 1536×864, and run the 10-trial reliability batch including the no-file and non-image cases. Record the comparison in the ExecPlan's Outcomes section.

Task 4: Expand the README into setup, usage, training and evaluation instructions, plus an architecture reference.

Task 5: Record the demo video. Carry-over from the poster.

Meetings with Client / Faculty Advisor

Date(s) of meeting(s) during Milestone 5: Oct 26th

Client / Faculty Advisor feedback on Milestone 5: Don't show both models' answers because a regular user without a tech background would not understand what that means. Opt to show only the more accurate result.

Faculty Advisor signature: Zahra Nematzadeh Date: 10/29/2020