

# BreastCancerDetectionCNN

## Milestone 4 Progress Evaluation

### 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 4 focused on the two technical challenges named in the project plan: the stalled held-out accuracy and the gap between validation and test scores.

**Challenge 1 (accuracy)** Lighter augmentation plus the SAM optimiser raised EfficientNet held-out F1 by about 0.07. Training at 512×288 instead of 224×224 raised held-out accuracy by about 0.04 at no extra training time. Applying the full recipe to ResNet-50 gave the project's best result.

- **Challenge 2 (validation vs. test gap) — root cause found and fixed.** 78–82% of every validation fold's images belonged to a patient who also had images in that fold's training data. Splitting folds by patient dropped EfficientNet's mean cross-validation F1 from 0.720 to 0.647, bringing validation in line with held-out results. ResNet-50 trained the same way shows validation F1 0.711 against held-out 0.660.
- **Semester success criterion met for both models.** ResNet-50 (5-fold ensemble, 512×288): held-out accuracy 0.713, F1 0.660, recall 0.710, against its baseline of 0.678 / 0.6534. EfficientNet-B0: F1 0.655 (single fold) and 0.647 (512×288 ensemble), against its baseline of 0.6375.

### Progress of Current Milestone (Progress Matrix)

| Task                                                                                       | Completion % | Kahlel | Woroma | Tara | To do                                                                |
|--------------------------------------------------------------------------------------------|--------------|--------|--------|------|----------------------------------------------------------------------|
| 1. Finish augmentation experiment retrain & evaluation (completed with an improved recipe) | 100%         | 100%   | 0%     | 0%   | Try higher resolutions and adjust augmentations to increase f1 score |
| 2. Implement & test app heatmap overlay (Grad-CAM)                                         | 100%         | 0%     | 100%   | 0%   | None                                                                 |
| 3. Web app polish (drag-and-drop, loading indicator, mobile) and CompareModels.py cleanup  | 90%          | 0%     | 80%    | 20%  | Finalize web app design                                              |
| Added: EfficientNet 3/5/10-fold comparison                                                 | 100%         | 100%   | 0%     | 0%   | None                                                                 |

|                                                        |      |      |     |    |                                         |
|--------------------------------------------------------|------|------|-----|----|-----------------------------------------|
| Added: EfficientNet light augmentation + SAM optimiser | 100% | 80%  | 20% | 0% | Isolate which of the two changes helped |
| Retrained both Models with 512×288 resolution.         | 100% | 100% | 0%  | 0% | Try a higher resolution                 |

## Discussion of Each Accomplished Task

### Task 1: Strengthened-augmentation experiment

The strengthened augmentation (random affine warp, wider brightness/contrast jitter, random erasing, plus MixUp) was evaluated for EfficientNet-B0 through the fold-count comparison below. The measured result was negative: 5-fold held-out F1 fell to 0.586 (single best fold) and 0.574 (ensemble), down from the 0.6375 baseline. Training loss never dropped below about 0.55, which showed the model was underfitting from too much regularisation. This result directly motivated the light-augmentation + SAM change.

### Added task: EfficientNet 3-, 5- and 10-fold comparison

A --folds option was added to EfficientNet.py and the model was trained with 3, 5 and 10 folds under identical settings. More folds helped on every held-out metric (10-fold single best F1 0.610 vs. 0.586 for 5-fold and 0.501 for 3-fold), but training time scaled accordingly (3h 51m, 6h 18m, 12h 27m). 5-fold was kept as the standard because 10-fold's gain did not justify doubling the run time.

### Added task: Light augmentation + SAM optimiser

Training augmentation was cut back to a horizontal flip and a 10° rotation, MixUp was turned off, and EfficientNet was switched to Sharpness-Aware Minimisation (SAM), an optimiser already used by the ResNet-50 model that favours flatter, better-generalising solutions. Held-out F1 rose to 0.655 (single best fold) and 0.646 (ensemble), about +0.07 over the heavy-augmentation run and above the 0.6375 baseline, with essentially no change in training time (about 6h 36m). Brightness and contrast jitter and random erasing were dropped because in a mammogram the local intensity and contrast of a lesion is the diagnostic signal, and an erased patch can remove the lesion entirely.

### Added task: Patient-grouped cross-validation and higher resolution

This task tested the plan's leading hypothesis for Challenge 2. The 2,864 training images come from 1,248 patients, with up to 24 images each, and the previous fold splitter ignored patients. Measured directly, ~80% of each validation fold's images belonged to patients also present in that fold's training data. Folds are now split with StratifiedGroupKFold by patient ID, verified to share no patient between train and validation for 3, 5 and 10 folds. Mean cross-validation F1 fell from 0.720 to 0.647 (run A), close to the held-out level, confirming the hypothesis.

Images were also being squashed from about 3,000×5,100 pixels to 224×224, distorting their shape by about 1.7×. Run B trained at 512×288, which preserves the aspect ratio. Compared with run A on the same patient-grouped folds, run B improved every metric: ensemble F1 +0.026, accuracy +0.038,

precision +0.051. It took no longer to train (about 6h 18m), because training is limited by JPEG decoding on the CPU, not by the GPU.

### Added task: ResNet-50 retrain and web app repair

ResNet-50 was retrained with the recipe that worked for EfficientNet: patient-grouped folds, light augmentation with MixUp off (it already used SAM), and 512×288 images. The 5-fold ensemble reached held-out accuracy 0.713, precision 0.616, recall 0.710 and F1 0.660.

The web app's inference module was updated so each model receives the uploaded image at the size it was trained on, and the app now serves the new ResNet ensemble. It uploads, returns a verdict, and a missing file or a non-image file shows the proper error message.

### Held-out results across the milestone (EfficientNet-B0)

All numbers are on the same 704 held-out test images with test-time augmentation (flips) and a 0.5 threshold. "Single best" is the fold with the highest validation F1, chosen without looking at the test set.

| Configuration                    | Mean CV F1 | Single best acc / F1 | Ensemble acc / prec / rec / F1 | Train time |
|----------------------------------|------------|----------------------|--------------------------------|------------|
| Plan baseline (Aug, single fold) | —          | 0.671 / 0.638        | —                              | —          |
| Heavy aug, 3-fold                | 0.527      | 0.618 / 0.501        | 0.655 / — / — / 0.539          | 3h 51m     |
| Heavy aug, 5-fold                | 0.544      | 0.635 / 0.586        | 0.658 / — / — / 0.574          | 6h 18m     |
| Heavy aug, 10-fold               | 0.546      | 0.646 / 0.610        | 0.663 / — / — / 0.584          | 12h 27m    |
| Light aug + SAM, 5-fold          | 0.720*     | 0.686 / 0.655        | 0.676 / 0.565 / 0.754 / 0.646  | 6h 36m     |
| + patient-grouped folds, 224×224 | 0.647      | 0.666 / 0.557        | 0.671 / 0.566 / 0.688 / 0.621  | 5h 50m     |
| + patient-grouped folds, 512×288 | 0.690      | 0.713 / 0.638        | 0.709 / 0.616 / 0.681 / 0.647  | 6h 18m     |

\* Inflated by patient leakage between training and validation folds; not comparable with the grouped runs.

### Held-out results (ResNet-50)

| Model                                                       | Threshold | Acc   | Prec  | Rec   | F1    |
|-------------------------------------------------------------|-----------|-------|-------|-------|-------|
| Plan baseline (Aug, 224×224, leaky folds; checkpoints lost) | 0.50      | 0.678 | —     | —     | 0.653 |
| Served by the app before repair (overwritten checkpoint)    | 0.45      | 0.587 | 0.480 | 0.656 | 0.554 |
| New: grouped folds, 512×288, best single fold               | 0.50      | 0.715 | 0.618 | 0.710 | 0.661 |
| New: grouped folds, 512×288, 5-fold ensemble                | 0.50      | 0.713 | 0.616 | 0.710 | 0.660 |

New ResNet: mean validation F1 0.711, training time 7h 17m.

### Task 2: Implemented Grad-CAM heatmap overlay

Wrote gradcam.py, hooking layer4 on ResNet-50 and features[-1] on EfficientNet-B0 to produce a Grad-CAM heatmap for whichever class each model actually voted for. Since both backbones are frozen, the input tensor requires its own gradient for backprop to reach the hooked layer; this was verified with a self-test on untrained weights before checkpoints were available, confirming the hooks and gradient flow work on both architectures independent of model quality. predict\_with\_heatmaps() was wired into app.py as a drop-in replacement for predict(), so a heatmap now renders under each model's verdict,

with a caption stating heatmaps show influence, not lesion outlines. The EfficientNet heatmap is the average of its five fold models; the same is now true for ResNet-50. Verified end-to-end against the updated inference pipeline and real checkpoints, including a case where the two models disagreed.

### Task 3: Web app polish + CompareModels.py cleanup

Removed the stale Distillation entry from CompareModels.py's PLOT\_DIRS and its subtitle, which referenced the nonexistent plots\_distill\_kfold output; verified the dashboard runs cleanly against only ResNet and EfficientNet.

Rebuilt the upload page with drag-and-drop, an image preview, and a loading indicator (an animated scan line plus a disabled submit button) so the interface signals progress during the model run. Verified the layout at both desktop and mobile widths. Redesigned the visual style for clarity.

### Evaluation against the semester success criteria

- **Accuracy / F1:** met for both models. ResNet-50: F1 0.660 vs. baseline 0.6534 and accuracy 0.713 vs. 0.678; the F1 gain alone is within noise, but accuracy clearly improved and recall stayed high (0.710). EfficientNet-B0: F1 0.655 / 0.647 vs. baseline 0.6375, accuracy up to 0.713 vs. 0.671. Cautions: with 704 test images, F1 differences under about 0.02–0.03 are within noise, and EfficientNet at 512×288 trades recall (0.68 vs. 0.75) for precision (0.62 vs. 0.57), meaning it misses more cancers at the default threshold.
- **Reproducibility:** met. All seven completed runs had empty error logs, the expected number of checkpoints (16,342,523 bytes each for EfficientNet, 107,226,339 bytes for ResNet) and the expected Excel fold-summary rows.
- **Speed:** partly measured. Model prediction for both models with flip averaging takes 0.25 seconds per image, full upload-to-page latency is scheduled for Milestone 5.
- **Reliability:** partly measured. An automated check of the app returned a verdict for a real mammogram and the correct error message for a missing file and a non-image file, the 10-trial batch is scheduled for Milestone 5.

### Obstacles

- Each full training run takes 4–12 hours on the single 8 GB GPU, so only one experiment can run at a time.
- 31 patients in the official CBIS-DDSM test set also appear in the training set. This slightly favours every model equally but messes with the held out test.

### Discussion of Each Team Member's Contribution

**Kahlel Cardona:** Worked on the training augmentation and fold comparison. In addition, grouped images by patients to not overlap training and validation tests. Ran the models with increased resolution.

**Woroma Dimkpa:** Built the Grad-CAM heatmap overlay for both architectures and integrated it into the live app; rebuilt the upload page with drag-and-drop, a loading indicator, and a mobile-responsive, clinically-styled interface; fixed the stale references in CompareModels.py.

**Taratong Dolinsky:** project's front-end development, including developing and maintaining the website used to showcase the project and beginning development of the portal login screen. She also creates project presentations and assists with reviewing and double-checking model training and results.

## Plan for Next Milestone (Milestone 5, Oct 26) — Task Matrix

| Task                                                                                                              | Kahlel | Woroma | Tara |
|-------------------------------------------------------------------------------------------------------------------|--------|--------|------|
| 1. Integrate the heatmap overlay into the live web app                                                            | 10%    | 80%    | 10%  |
| 2. Full evaluation suite: held-out accuracy/F1 for both models, upload-to-result latency, reliability trial batch | 100%   | 0%     | 0%   |
| 3. Senior Design Showcase poster                                                                                  | 5%     | 5%     | 90%  |
| 4. Make the Final Answer respect each model's threshold; try a higher training resolution                         | 60%    | 40%    | 0%   |
| 5. Carry-over: finish any remaining Milestone 4 Task 2/3 work                                                     | 30%    | 0%     | 70%  |

### Discussion of Each Planned Task for Milestone 5

**Task 1:** Show the heatmap overlay next to each model's verdict. The app already serves the new ResNet and can feed each model its own image size, so switching EfficientNet to the 512×288 ensemble is a one-line change in eval\_results/selected\_models.json. Before switching, the team should decide whether its lower recall is acceptable, or choose a decision threshold on validation data (never the test set) to recover it.

**Task 2:** Run evaluate.py on both architectures' final checkpoints, measure end-to-end web-app latency (target under 2 seconds per image), and run the 10-trial reliability batch including the no-file and non-image edge cases.

**Task 3:** Build the showcase poster covering the system architecture, the patient-leakage finding, and the semester's before/after results.

**Task 4:** Change how the app blends the two models so the Final Answer can no longer contradict two agreeing verdicts. Separately, try a higher training resolution (e.g. 768×432): at 512×288 training time was set by image loading, not the GPU, so higher resolution may be cheap.

**Task 5:** Complete any remaining Milestone 4 development, including finishing the Grad-CAM heatmap integration and continuing web application improvements. This will also include further front-end development of the project portal, building on the login screen already developed, along with any remaining interface and code cleanup.

## Meetings with Client / Faculty Advisor

Date(s) of meeting(s) during Milestone 4: 9/29

Faculty Advisor Evaluation

Score each member from 0 to 10 for Milestone 4.

| Member             | Score (0-10) |
|--------------------|--------------|
| Kahlel Cardona     |              |
| Woroma Dimkpa      |              |
| Taratong Dollinsky |              |

Faculty Advisor signature: Zahne Menzies Date: 9/29/2026