

PROJECT ABSTRACT

Breast Cancer Detection using CNN

CNN classifier for breast histopathology patches (benign vs malignant) on the BreakHis dataset, with stain normalization and patient-wise evaluation, delivered as a built-to-order student project with code, training notebook, Flask demo, report, PPT and viva kit.

01. Title

Breast Cancer Detection using CNN

02. Introduction / Background

Breast cancer diagnosis rests on the pathologist's microscope: H&E-stained tissue slides are examined at multiple magnifications for the architectural and nuclear changes that separate benign from malignant tissue. It is skilled, time-consuming work, and pathologist workloads keep growing — which is why automated histopathology analysis is one of the most published areas in medical computer vision. The BreakHis dataset (Spanhol et al., 2016) — 7,909 microscopy images from 82 patients, labeled benign or malignant across 8 subtypes and 4 magnifications — is the standard academic benchmark. This project builds the canonical student solution: a transfer-learned patch classifier with stain normalization and strict patient-wise splitting.

03. Problem Statement

Patches from the same patient look nearly identical, so an image-wise split leaks near-duplicates into the test set and inflates accuracy — the classic leakage trap in histopathology projects. H&E stain intensity also varies between labs, shifting image colors. A defensible build must split by patient ID, normalize stains, and report per-magnification metrics (40x–400x are different visual tasks). The project addresses this with a leakage-safe patch pipeline and open subtype error analysis, while being explicit that it is an academic research prototype — histopathology diagnosis stays with the pathologist.

04. Objectives

- Fine-tune ResNet50 / EfficientNet-B0 (ImageNet transfer learning) on BreakHis patches for binary benign vs malignant classification, with the 8-subtype experiment as a documented extension.

- Reach design targets of ~0.90 binary accuracy and ~0.93 AUC, with the notebook logging accuracy, AUC, sensitivity, specificity and the confusion matrix per magnification on held-out patients.
- Apply Macenko-style H&E stain normalization with a documented with/without comparison, since stain intensity varies between labs.
- Build a Flask demo app with patch upload, magnification switching, nuclei-highlight overlay and a benign-vs-malignant comparison viewer; deliver the full student kit (code, notebook, report, PPT, viva Q&A).

05. Proposed System / Working

The system has three stages. (1) Data pipeline: BreakHis images (7,909 H&E patches, 700x460 px, benign vs malignant with 8 subtype labels, 40x–400x) are grouped by patient ID; patients are split into train/validation/test sets so no leakage occurs. Patches get H&E stain normalization, resize to 224x224 and microscopy-appropriate augmentation. (2) Model: an ImageNet-pretrained ResNet50 (or EfficientNet-B0) is fine-tuned with a binary head, logging loss, accuracy and AUC per epoch on the validation patients. (3) Demo application: a Flask app classifies an uploaded patch, shows benign/malignant probabilities with the nuclei-highlight overlay, and offers the side-by-side benign-vs-malignant comparison viewer.

06. Technologies / Components Used

- Python 3.10, PyTorch with timm (ResNet50 / EfficientNet-B0 transfer learning)
- OpenCV, scikit-image (patch extraction, stain normalization)
- scikit-learn (metrics, ROC-AUC, patient-wise splits)
- Matplotlib, Seaborn (curves, ROC plots, confusion matrix)
- Flask demo web app (patch upload, comparison viewer, nuclei overlay)
- BreakHis dataset (Spanhol et al., 2016; 7,909 images, 82 patients)
- Jupyter training and evaluation notebook

07. Key Features

- Binary histopathology patch classification: benign vs malignant
- Patient-wise split enforcement — patches from one patient never split across sets
- H&E stain normalization with documented with/without comparison
- Multi-magnification training (40x/100x/200x/400x) with per-magnification reporting
- Nuclei-highlight viewer visualizing nuclear crowding and irregularity
- Subtype error analysis across the 8 BreakHis subtypes
- Patch extraction pipeline with background filtering and configurable stride

08. Applications / Use Cases

- Teaching material for histopathology deep learning: patch pipelines, stain normalization, patient-wise evaluation, AUC
- Reference build for other microscopy classification tasks
- Demo of rigorous evaluation practice in a leakage-prone domain
- Prototype scope only: a classroom/research demonstration, never clinical use — histopathology diagnosis stays with the pathologist

09. Results & Scope

The deliverable is a design-target-driven build, not a pre-measured product. The training notebook computes accuracy, AUC, sensitivity, specificity and the confusion matrix — per magnification and per subtype — on held-out patients; the report presents these as the build's measured results. Design targets are ~0.90 binary accuracy and ~0.93 AUC. Scope is patch-level classification from a single lab's research dataset — it is not whole-slide diagnosis, does not generalize to other labs unpromised, and is strictly an academic prototype, not a medical device.

10. Conclusion

Breast Cancer Detection using CNN delivers an end-to-end histopathology project: a real diagnostic problem, the standard BreaKHis benchmark, a transfer-learned ResNet trained through a fully logged experiment with leakage-safe patient-wise splits and stain normalization, and a demo app that makes the result visible in seconds. The honest framing — design targets, per-magnification metrics, open subtype error analysis — gives the student genuine, defensible material for the report and viva.

11. References

- Spanhol, F. A. et al. — A Dataset for Breast Cancer Histopathological Image Classification, IEEE Transactions on Biomedical Engineering, 2016 (BreaKHis).
- He, K. et al. — Deep Residual Learning for Image Recognition (arXiv:1512.03385), the ResNet backbone family used via transfer learning.
- Macenko, M. et al. — A method for normalizing histology slides for quantitative analysis, IEEE ISBI 2009 (stain normalization).
- PyTorch and timm documentation — training and transfer-learning tooling used in the build.