Hybrid Deep Learning for Breast Cancer Histopathology: Phased Fine-Tuning, Cross-Dataset Generalization with Adaptive Normalization, and Web Deployment

Yousef Khaled , Abdelrahman Ahmed , Belal Gamal , Menna Mohamed

Department of Communication and Information Engineering, Zewail City of Science & Technology

CIE 555 – CIE 552

Professor Omar M. Fahmy – Professor Eman Badr

May 11, 2025

Abstract

Breast cancer remains a significant global health concern, and accurate, timely diagnosis from histopathological images is crucial for effective treatment. Deep learning models have shown considerable promise in this domain. This paper presents the development, evaluation, and deployment of a hybrid deep learning architecture, combining features from EfficientNetV2-S and a Vision Transformer (ViT-B/16), for the binary classification of breast cancer histology images (benign vs. malignant) from the BreaKHis dataset. We detail a phased fine-tuning strategy designed to optimize performance by progressively unfreezing backbone layers. Furthermore, we explore the impact of advanced computer vision techniques, including stain normalization using global dataset statistics and Test-Time Augmentation (TTA), to assess model robustness and generalization on an external dataset (BACH). The methodology includes rigorous data preprocessing, handling class imbalance, and patient-level data splitting. The BreaKHis-trained model achieved a test Area Under the Curve (AUC) of 0.8048, and training (AUC) of 8900. When tested on a quality-controlled subset of the BACH dataset with domain-adaptive normalization and TTA, an AUC of 0.6811 was observed, highlighting challenges in domain generalization. Finally, the best-performing model was deployed as an interactive web application using Gradio and Hugging Face Spaces, demonstrating a pathway from research to practical application. Future work will focus on multi-class classification and advanced meta-learning approaches.

Keywords: Breast Cancer, Histopathology, Deep Learning, Hybrid Model, EfficientNet, Vision Transformer, Phased Fine-Tuning, Stain Normalization, Test-Time Augmentation, Domain Generalization, Model Deployment, Gradio, Hugging Face Spaces.

1. Introduction

Breast cancer is one of the most prevalent cancers worldwide, with histopathological examination of tissue biopsies serving as the gold standard for diagnosis *[Sung et al., 2021; Gurcan et al., 2009]*. The interpretation of these images by pathologists is a complex task, often time-consuming and subject to inter-observer variability *[Elmore et al., 2015]*. Computer-Aided Diagnosis (CADx) systems, particularly those leveraging deep learning, have emerged as powerful tools to assist pathologists by providing objective and potentially faster analyses *[Litjens et al., 2017]*.

Convolutional Neural Networks (CNNs) have demonstrated remarkable success in medical image analysis. More recently, Vision Transformers (ViTs) *[Dosovitskiy et al., 2020]* have shown competitive performance by applying the transformer architecture to image patches, effectively capturing global context. Hybrid models that combine the strengths of CNNs

(strong local feature extraction) and ViTs (global context modeling) are a promising avenue for further improving performance in complex visual tasks like histopathology *[Chen et al., 2021; Zarif et al., 2024]*.

The BreaKHis dataset *[Spanhol et al., 2016]* is a widely used public benchmark for breast cancer histopathology. However, it presents challenges such as class imbalance and stain variability. This study aims to develop a robust hybrid deep learning model for binary breast cancer classification (benign vs. malignant) on the BreaKHis dataset and evaluate its generalization. Our contributions include:

- The design and implementation of a hybrid architecture fusing features from EfficientNetV2-S and ViT-B/16.
- A systematic phased fine-tuning approach with differential learning rates.
- Exploration of Reinhard stain normalization targeted towards global BreaKHis training set statistics for testing on an external dataset (BACH challenge data *[Aresta et al., 2019]*).
- Application of Test-Time Augmentation (TTA) to improve predictive stability.
- Rigorous evaluation methodology, including patient-level data splitting.
- Deployment of the trained model as an interactive web application using Gradio and Hugging Face Spaces, demonstrating practical applicability.

2. Related Work

The application of deep learning to breast cancer histopathology has rapidly evolved, moving from foundational Convolutional Neural Network (CNN) architectures to more sophisticated models and techniques. Early efforts often focused on adapting well-known CNNs like AlexNet, VGG, and ResNet for patch-based classification on datasets such as BreaKHis [Spanhol, F. A., Oliveira, L. S., Petitjean, C., & Heutte, L. (2016)]. Breast Cancer Histopathological Image Classification Using Convolutional Neural Networks. International Joint Conference on Neural Networks (IJCNN), 2560-2567.]. Spanhol et al. (2016) themselves provided initial benchmarks on BreaKHis using CNNs, demonstrating their superiority over traditional texture-based descriptors like LBP and GLCM, which they also evaluated in their work introducing the dataset [Spanhol, F. A., Oliveira, L. S., Petitjean, C., & Heutte, L. (2016). A Dataset for Breast Cancer Histopathological Image Classification. IEEE Transactions on Biomedical Engineering, 63(7), 1455–1462.].

More recent research has explored hybrid architectures to leverage diverse feature learning capabilities.

For instance, Zarif et al. (2024) proposed a hybrid model fusing a custom CNN with EfficientNetV2B3 for classifying invasive ductal carcinoma, highlighting the benefits of combining local feature extraction with deeper, pre-trained representations [Zarif, M. I., Shahin, I., Almotiri, J., & Othman, M. (2024). Using Hybrid Pre-trained Models for Breast Cancer Detection. PLOS ONE, 19(1), e0296912.]. Our work extends this hybrid paradigm by integrating EfficientNetV2-S with a Vision Transformer (ViT) [Dosovitskiy et al., 2020], aiming to capture both strong local features and global image context effectively, a combination increasingly recognized for its potential in complex medical image analysis.

A significant challenge in histopathology, particularly with datasets like BreaKHis, is the variability in image magnification. Bayramoglu et al. (2016) addressed this by developing a magnification-independent CNN approach for BreaKHis, demonstrating the feasibility of models that can generalize across different zoom levels without explicit magnification input during inference [Bayramoglu, N., Kannala, J., & Heikkilä, J. (2016). Deep Learning for Magnification Independent Breast Cancer Histopathology Image Classification. 23rd International Conference on Pattern Recognition (ICPR), 2440–2445.]. While our current model does not explicitly embed magnification information, our phased fine-tuning on images from all magnifications aims for robustness to this factor.

Stain variability is another critical hurdle for generalization. Techniques such as Reinhard normalization [Reinhard et al., 2001], which we employ, and stain deconvolution methods like Macenko et al. [Macenko et al., 2009], are commonly used. Araujo et al. (2017), in their work on the BACH dataset (derived from the ICIAR 2018 Challenge, also used as our external test set), utilized the Macenko method for stain normalization before classifying patches into four categories (normal, benign, in situ, invasive) using a CNN, later combining patch predictions for image-level decisions [Araujo, T., Aresta, G., Castro, E., Rouco, J., Aguiar, P., Eloy, C., ... & Campilho, A. (2017). Classification of Breast Cancer Histology Images Using Convolutional Neural Networks. PLOS ONE, 12(6), e0177544. See also Aresta, G., Araújo, T., Kwok, S., et al. (2019). Bach: Grand challenge on breast cancer histology images. Medical image analysis, 56, 122-139.]. This underscores the importance of stain normalization when dealing with multi-source or external datasets.

Fine-tuning strategies, including phased unfreezing and differential learning rates as used in our study, are crucial for adapting large models pre-trained on natural images (e.g., ImageNet) to specialized medical imaging tasks without catastrophic forgetting and while leveraging learned features effectively.

Finally, the challenge of limited annotated data in medical imaging has spurred interest in few-shot learning. Li et al. (2024) explored few-shot learning based on VGG16 with contrastive loss for colorectal cancer histopathology, demonstrating high accuracy with

minimal training samples [Li, R., Li, X., Sun, H., Yang, J., Rahaman, M., Grzegozek, M., ... & Li, C. (2024). Few-shot Learning Based Histopathological Image Classification of Colorectal Cancer. Intelligent Medicine, 4, 256-267.]. This direction aligns with our future work considerations for meta-learning approaches to enhance domain adaptation and generalization with limited data. Our study builds upon these foundations by developing a novel hybrid architecture with a systematic training and evaluation methodology, explicitly addressing domain adaptation through stain normalization for external dataset testing.

3. Methodology

3.1. Dataset Description

3.1.1. BreakHis Dataset

The primary dataset used for training and initial testing is the BreaKHis (Breast Cancer Histopathology Image Classification) dataset *[Spanhol et al., 2016]*. It contains 7,909 microscopic images of breast tumor tissue, categorized into benign and malignant classes. The images were acquired at four different magnification factors: 40X, 100X, 200X, and 400X. Each image is 700x460 pixels in 3-channel RGB PNG format. The dataset further subdivides benign tumors into adenosis (A), fibroadenoma (F), phyllodes tumor (PT), and tubular adenoma (TA), and malignant tumors into ductal carcinoma (DC), lobular carcinoma (LC), mucinous carcinoma (MC), and papillary carcinoma (PC).
For this study, we focus on the binary classification task (benign vs. malignant).
Each image is associated with a slide ID, which is crucial for patient-level data splitting.

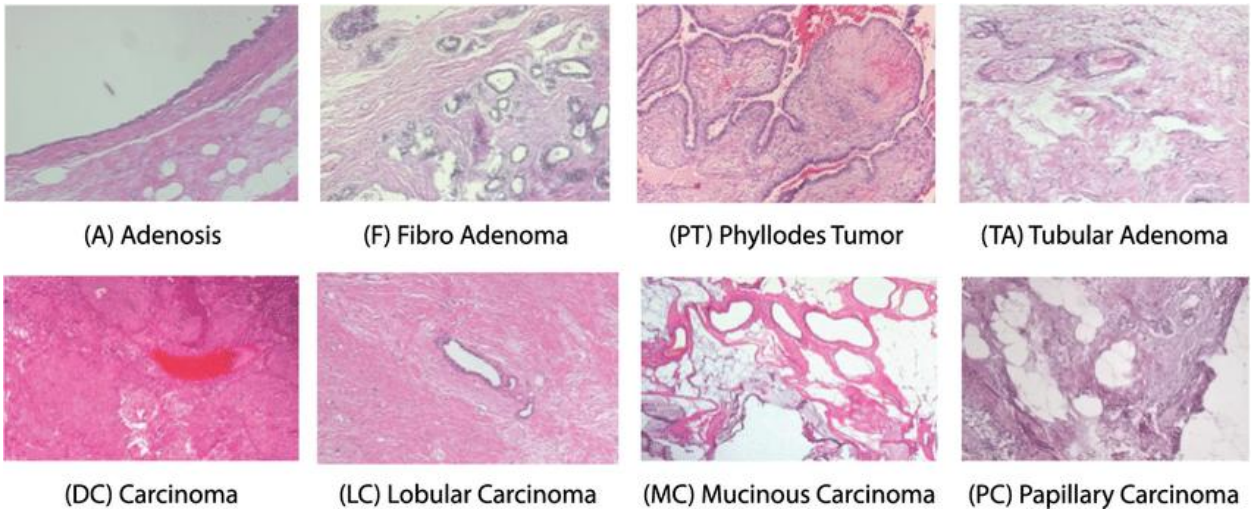

Figure 1

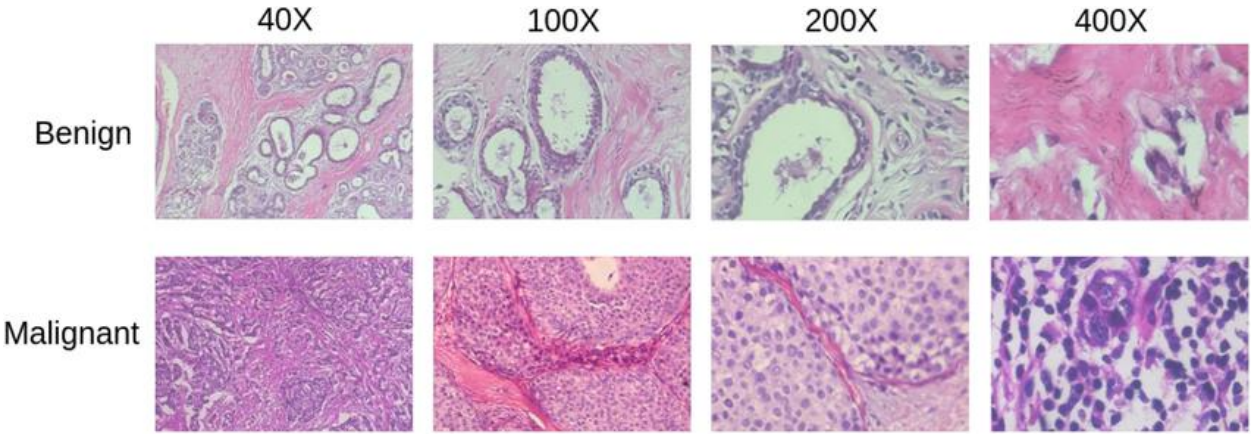

Figure 2

3.1.2. External Test Dataset (BACH - Breast Cancer Cell Histopathological Images)

To assess the generalization capability of our trained model, we utilized the "Breast Cancer Cell (Histopathological images)" dataset from Kaggle, originally derived from the ICIAR 2018 BACH Challenge *[Aresta et al., 2019]*. This dataset contains 440 image patches labeled as Benign, In Situ Carcinoma, Invasive Carcinoma, or Normal. For our binary task, "Normal" and "Benign" were mapped to our "benign" class, while "In Situ Carcinoma" and "Invasive Carcinoma" were mapped to our "malignant" class.

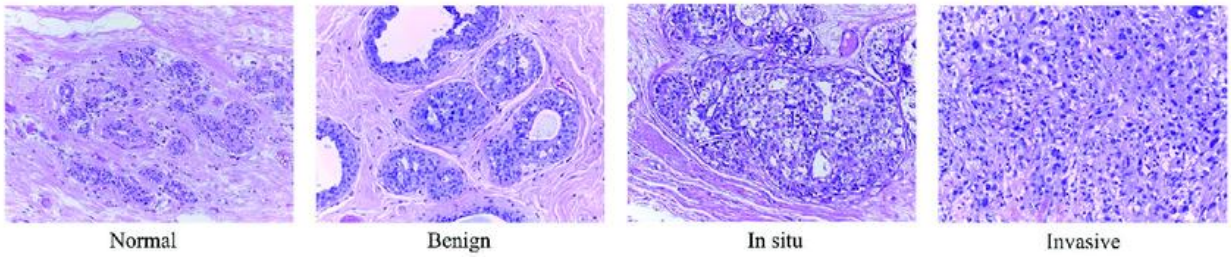

Figure 3

3.2. Data Preprocessing and Organization

3.2.1. BreakHis Data Organization

The BreaKHis dataset was initially downloaded and unzipped. Images were reorganized from their original complex directory structure into two main folders: /content/organized_images/benign and /content/organized_images/malignant. A metadata CSV file was created, parsing filenames to extract image path, binary label (benign/malignant), tumor subtype, magnification level, and patient slide ID. This patient slide ID was used for group-based data splitting.

3.2.2. Label Encoding

A sklearn.preprocessing.LabelEncoder was used to convert the string labels ('benign', 'malignant') into numerical format (e.g., 0 and 1) for model training.

3.2.3. Data Splitting

To prevent data leakage and ensure a robust evaluation, the BreaKHis dataset was split into training, validation, and test sets using sklearn.model_selection.GroupShuffleSplit based on patient slide_id. This ensures that all images from a single patient slide belong exclusively to one set. The dataset was first split to create a test set comprising 15% of the total data. The remaining 85% (for training and validation) was then further split, allocating approximately 15% of the original total data to the validation set. This resulted in final approximate proportions of 70% for the training set, 15% for the validation set, and 15% for the test set.

3.3. Model Architecture: Hybrid CNN-Transformer

We developed a hybrid model (HybridModel) designed to synergistically combine the distinct feature extraction capabilities of Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs). The rationale is that CNNs excel at learning hierarchical local features and spatial invariances, while ViTs are proficient at capturing global relationships and long-range dependencies within an image.

Backbones:

- **EfficientNetV2-S:** Chosen as the CNN backbone due to its balance of high accuracy and computational efficiency, achieved through its Fused-MBConv blocks and progressive learning strategy *[Tan & Le, 2021]*. It was pre-trained on ImageNet (IMAGENET1K_V1 weights) to leverage rich, general-purpose visual features. Its feature extractor component (model.features) was utilized to obtain multi-scale feature maps.
- **Vision Transformer (ViT-B/16):** Selected as the Transformer backbone for its demonstrated strength in modeling global image context by treating image patches as sequences *[Dosovitskiy et al., 2020]*. It was also pre-trained on ImageNet (IMAGENET1K_V1 weights). To adapt it for feature extraction, its original classification head was removed (model.heads = nn.Identity()). Features were extracted from the output corresponding to the [CLS] token after the Transformer encoder, as this token is designed to aggregate global image information.

Feature Processing:

- **EfficientNetV2-S Features**: The output feature maps from EfficientNetV2-S (typically having spatial dimensions) were passed through an

nn.AdaptiveAvgPool2d((1, 1)) layer. This global average pooling reduces each feature map to a single vector per channel, effectively summarizing the spatial information for each feature before being flattened into a 1D feature vector. This step ensures a fixed-size output regardless of minor variations in input image size to the backbone and provides a compact representation of the CNN's learned features.

- **ViT-B/16 Features**: The [CLS] token output from the ViT encoder, which is already a 1D feature vector representing the global image summary, was used directly.

Fusion Block:

The flattened features from both backbones were concatenated. This combined feature vector was then processed by a fusion block consisting of:
LayerNorm -> Linear(input_dim, 1024) -> GELU -> LayerNorm(1024) -> Dropout(0.6) -> Linear(1024, 1024)

Classifier Head:

The output of the fusion block was passed to a classifier head:
LayerNorm(1024) -> Linear(1024, 512) -> GELU -> Dropout(0.7) -> Linear(512, num_classes)
where num_classes is 2 for binary classification.

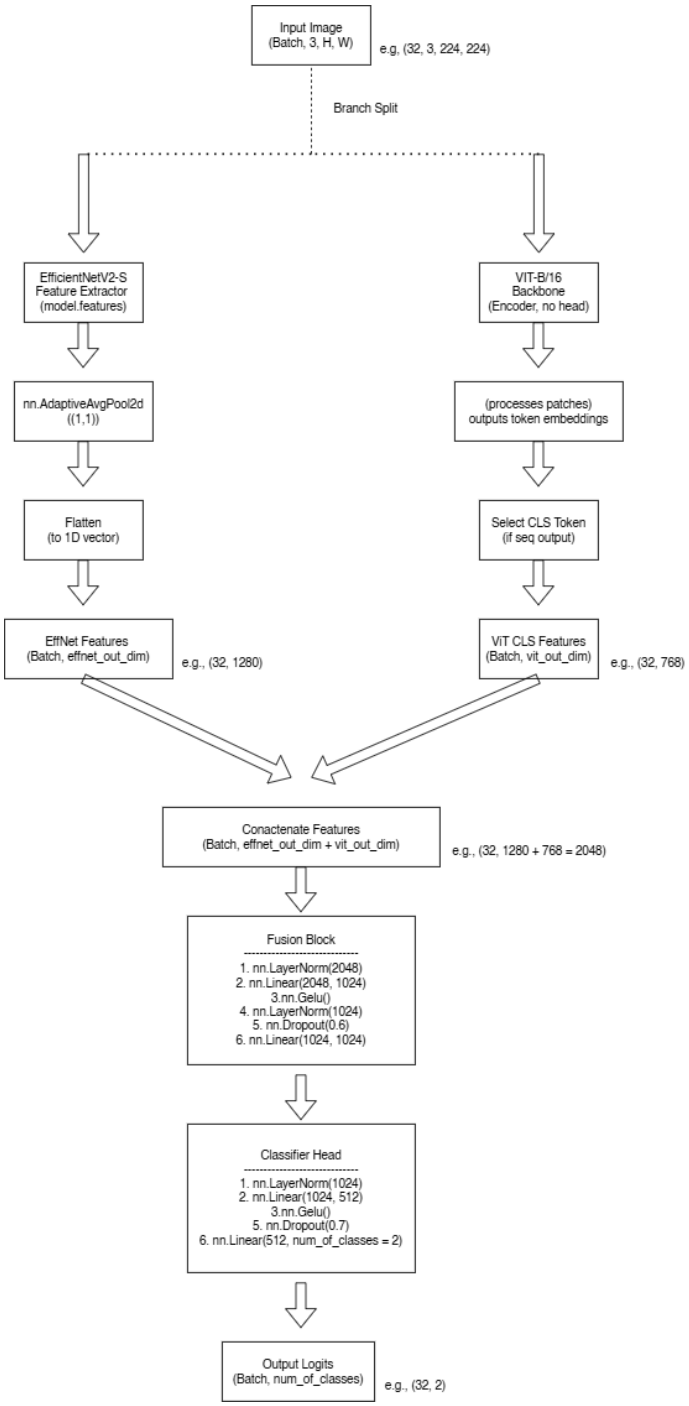

Figure 4

3.4. Training Strategy

3.4.1. Loss Function

A nn.CrossEntropyLoss was employed. To address class imbalance in the BreaKHis dataset, class weights were computed using

sklearn.utils.class_weight.compute_class_weight('balanced', ...) and applied to the loss function. Label smoothing with a factor of 0.2 was also used to regularize the model and prevent overconfidence.

3.4.2. Optimizer

The torch.optim.AdamW optimizer was used with a base weight decay of 1e-5.

3.4.3. Phased Fine-Tuning and Learning Rates

A systematic three-phase fine-tuning strategy was implemented to effectively adapt the pre-trained backbones (EfficientNetV2-S and ViT-B/16) to the specific domain of breast cancer histopathology while mitigating the risk of catastrophic forgetting and optimizing computational resources. This progressive unfreezing approach allows the model to learn task-specific features in a hierarchical manner, starting from the higher-level, randomly initialized layers and gradually fine-tuning deeper, pre-trained layers. Differential learning rates were employed to provide finer control over the update magnitudes for different parts of the network.

- **Phase 1: Head and Fusion Block Training:**
    - **Action:** Only the randomly initialized fusion block and the final classifier head were trained. The pre-trained backbones (EfficientNetV2-S features and ViT encoder) remained frozen.
    - **Rationale:** This initial phase allows the newly added, task-specific layers to adapt to the robust, general-purpose features extracted by the frozen pre-trained backbones. Training only these smaller, higher-level parts of the network is computationally efficient and helps establish a stable baseline before disturbing the well-learned weights of the backbones. A relatively higher learning rate (e.g., 1e-4) was used for the head components, as these parameters are starting from scratch.
- **Phase 2: Partial Backbone Unfreezing and Fine-Tuning:**
    - **Action:** In addition to the head and fusion block, later layers of both backbones were unfrozen. For instance, the last 3 stages of EfficientNetV2-S and the last 4 encoder blocks (along with associated projection/token layers) of ViT-B/16 were made trainable.
    - **Rationale:** Features in deeper layers of pre-trained models are often more specialized to the original training domain (e.g., ImageNet). By unfreezing only the later layers, we allow the model to adapt these more specialized features to the nuances of histopathological images without drastically altering the more general, foundational features learned in earlier layers. This

targeted fine-tuning helps to refine representations for the specific task. Significantly smaller learning rates were used for these newly unfrozen backbone parts (e.g., 1e-6) compared to the head (e.g., 2e-5) to ensure gentle adaptation and prevent the destruction of valuable pre-trained knowledge.

- **Phase 3: Full Backbone Fine-Tuning:**
  - **Action:** All model parameters, including all layers of both backbones, were unfrozen for end-to-end fine-tuning.
  - **Rationale:** This phase aims to allow the entire network to adapt holistically to the target dataset, potentially capturing subtle domain-specific characteristics throughout the feature hierarchy. However, this phase requires the most care due to the large number of trainable parameters. Even smaller differential learning rates were applied (e.g., head 1e-5, backbone 5e-7 or similar) to maintain stability and allow for fine-grained adjustments across the entire model. This phase is particularly sensitive to overfitting if learning rates are not sufficiently small or if the dataset size is limited relative to the model's capacity.

Learning rate adjustments between phases and the decision to transition from one phase to the next were guided by observing the validation AUC, training stability (e.g., loss curves, gradient magnitudes), and the number of epochs allocated per phase. The goal was to maximize performance on the validation set while ensuring a stable learning process.

3.4.4. Learning Rate Scheduler

A torch.optim.lr_scheduler.ReduceLROnPlateau scheduler was used, monitoring the validation AUC. If validation AUC did not improve for a patience of 3 epochs, the learning rate was reduced by a factor of 0.2.

3.4.5. Data Augmentation

Standard data augmentations were applied during training using torchvision.transforms, including:

Resize (to 256x256) - RandomCrop (to 224x224) – RandomHorizontalFlip - RandomVerticalFlip - ColorJitter (brightness, contrast, saturation, hue) – RandomRotation - RandomAffine (translation) – RandomErasing.

The intensity of these augmentations was varied per phase, with stronger augmentations in early phases and milder ones later.
Test images were resized to 224x224 and normalized.

3.4.6. Mixed Precision Training

torch.amp (Automatic Mixed Precision) was used with a GradScaler to accelerate training and reduce memory usage, where feasible. Gradient clipping (max norm 1.0) was applied after unscaling gradients and before the optimizer step, especially during phases with unfrozen backbones.

3.4.7. Handling Class Imbalance (Sampler)

For the training DataLoader, a WeightedRandomSampler was implemented to oversample the minority class, ensuring each batch had a more balanced representation of classes.

3.5. Evaluation Metrics

Model performance was evaluated using: Accuracy, Area Under the ROC Curve (AUC), Precision, Recall, and F1-Score. A confusion matrix and classification report were also generated. The primary metric for model selection and early stopping was Validation AUC.

3.6. Test-Time Augmentation (TTA) on External Dataset

For evaluating on the external BACH dataset, TTA was implemented. Multiple augmented versions of each test image (e.g., original, horizontal flip, vertical flip, 90/180/270 rotations) were generated. Predictions (probabilities) for each augmented version were averaged to produce the final prediction for the image.

3.7. Stain Normalization for External Dataset Testing

To adapt the external BACH dataset to the BreaKHis domain, Reinhard stain normalization was applied to BACH images. Crucially, the target statistics (mean and standard deviation in LAB color space) for this normalization were derived from the entire BreaKHis training set (or a large representative sample thereof using an online calculation method to manage memory). This aims to make BACH images appear as if they were stained similarly to the average BreaKHis training image. This normalization was applied before TTA and other standard test transforms.

4. Experiments and Results

4.1. Experimental Setup

- Hardware: Google Colab (T4 GPU)
- Software: PyTorch, Python, and key libraries.
- Training Hyperparameters:
    - Batch Size: 32
    - Total Max Epochs: 30

- Phase Epochs: Phase 1: 7, Phase 2: 10, Phase 3: 13
- Optimizer: AdamW, Weight Decay 1e-5
- Initial Learning Rates: (as described in 3.4.3)
- Early Stopping Patience: 7 epochs based on Val AUC.

4.2. Performance on BreaKHis Dataset

Best Overall Model Performance (Epoch 9, Phase 2):

Metric	Value of Training Set	Value of Validation Set	Value of Test Set
Accuracy	94.19%	84.64%	79.07%
Loss	0.3617	0.7391	---
Precision	---	0.9659	0.7847
Recall	---	0.8567	0.8496
F1-Score	---	0.9080	0.8159
AUC	---	0.9199	0.8048

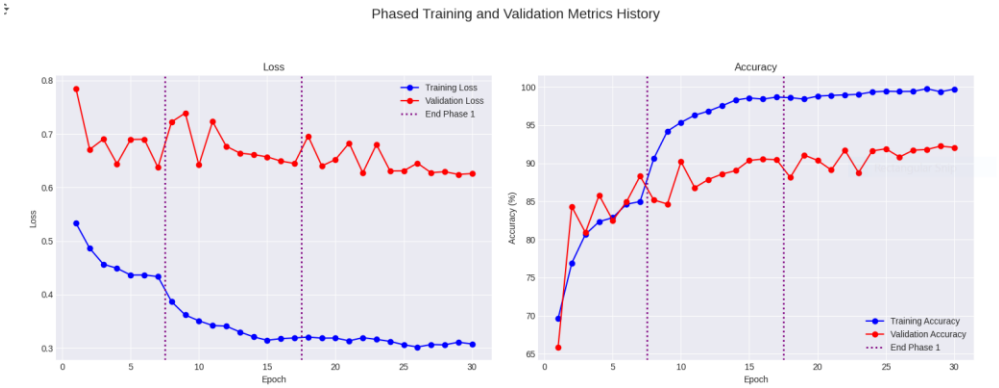

Figure 5

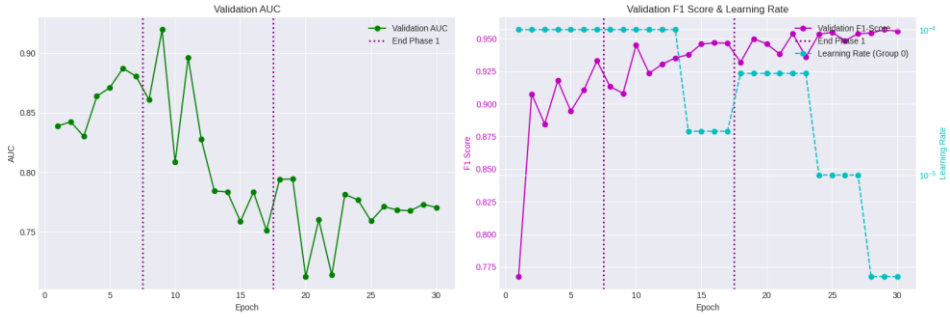

Figure 6

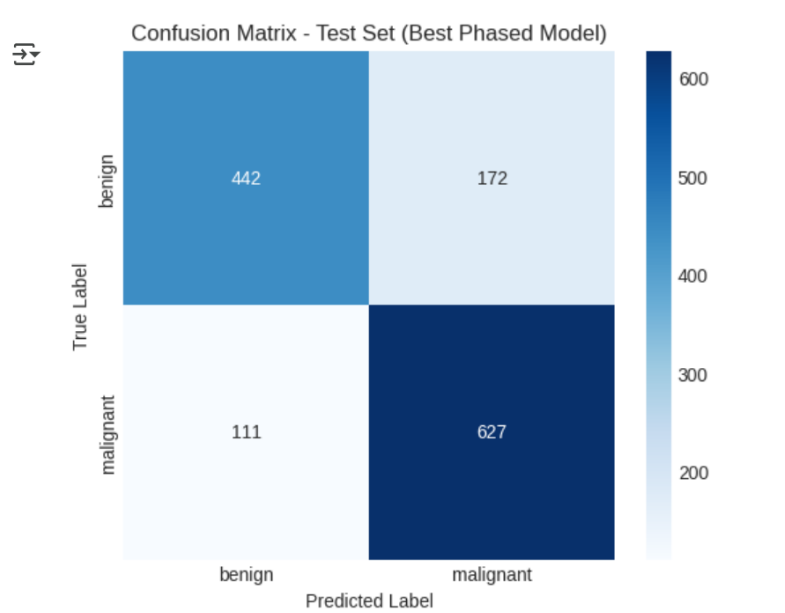

Figure 7

How well did the model perform on BreaKHis?

On the **validation set**, the model achieved a peak AUC of **0.9199**. The corresponding validation accuracy was 84.64%, precision 0.9659, recall 0.8567, and F1-score 0.9080.

The performance on the actual **test set** (which you provided separately with the confusion matrix) was an AUC of 0.8048 and an accuracy of 79.07%. This indicates a notable drop from the best validation performance, suggesting some overfitting to the validation set or that the validation set was slightly easier than the test set

At which phase/epoch was the best performance achieved?

The best performance (highest validation AUC of 0.9199) was achieved during **Phase 2, at Epoch 9.**

Did the phased fine-tuning strategy seem effective? (e.g., did performance improve when backbones were unfrozen, or was head training already optimal?)

The phased fine-tuning strategy appeared effective up to Phase 2.

Phase 1 established a decent baseline, with the model learning from the head and fusion block (max Val AUC 0.8874).

Phase 2, where parts of the backbones were unfrozen, led to a significant improvement in validation AUC, jumping to 0.9199. This clearly indicates that fine-tuning some pre-trained backbone layers was beneficial and improved upon what the head alone could achieve.

Phase 3, where the entire backbones were unfrozen, did **not** lead to further improvement in the primary metric (validation AUC). In fact, the validation AUC generally decreased or stagnated during this phase. This suggests that either:

The learning rates in Phase 3 were still too high for full fine-tuning, leading to instability or overfitting.

The model had already reached its peak capacity with the partially unfrozen backbones in Phase 2, and unfreezing more parameters introduced too much noise or complexity for the given data and learning rate.

The dataset size might not have been sufficient to effectively fine-tune such a large number of parameters (over 100 million) without overfitting.

Observations on training stability, convergence.

Training Stability:

Training loss consistently decreased, and training accuracy consistently increased across all phases, reaching near-perfect scores. This indicates the model was able to fit the training data very well.

Validation loss was more erratic. It generally decreased in Phase 1, then showed some fluctuations in Phase 2 and 3, sometimes increasing even when AUC was high. This is common, as loss and AUC are not always perfectly correlated.

The gradient monitoring showed that gradients were present in the unfrozen backbone parts during Phases 2 and 3, indicating that these layers were indeed receiving learning signals. The magnitudes seemed reasonable (not excessively large or vanishingly small).

Convergence:

The model appeared to converge to its best validation performance (based on AUC) relatively early in Phase 2 (Epoch 9).

While training accuracy continued to climb, the validation AUC did not improve further after this point, suggesting that the model started to overfit the training data or that the validation set was not representative enough to guide further useful generalization.

The learning rate scheduler appropriately reduced learning rates when validation AUC plateaued, but this did not lead to new peaks in performance after Epoch 9.

The fact that the early stopping condition was met repeatedly from Epoch 16 onwards (but disabled) strongly suggests that continuing training for the full 30 epochs beyond this point was not beneficial for generalization on the validation set metric.

4.3. Generalization to External BACH Dataset (dina0808/bach-icar-2018)

To assess the generalization capabilities of the BreaKHis-trained model, it was evaluated on the "BACH (ICIAR 2018) Part 1" dataset sourced from Kaggle (dina0808/bach-icar-2018). This dataset consists of 400 image patches **(2048x1536** pixels, TIFF format) categorized into Normal, Benign, In Situ carcinoma, and Invasive carcinoma.

4.3.1. BACH Dataset Preprocessing

1. Label Mapping: The four original BACH classes were mapped to binary labels ('benign', 'malignant') consistent with the BreaKHis task: 'Normal' and 'Benign' (BACH) were mapped to 'benign'; 'In Situ carcinoma' and 'Invasive carcinoma' (BACH) were mapped to 'malignant'.
2. Quality Control (QC): To ensure a higher quality test set from BACH, a QC pipeline was applied:
    - Blur Detection: Laplacian variance was calculated for each image. Images with a variance below a threshold of 75.0 were discarded as too blurry.
    - Tissue Area Detection: A simple intensity-based thresholding (bg_threshold_value = 220) was used to estimate the percentage of non-background (tissue) pixels. Images with less than 40.0% tissue content were discarded.
    - Out of an initial 400 images in the dataset, 275 images passed these QC criteria.
3. Stain Normalization: The QC-passed BACH images were stain-normalized using the Reinhard method. The target statistics (mean and standard deviation in LAB color space) for this normalization were derived from the global distribution of the BreaKHis training set (using a sample of 2000 BreaKHis images via an online calculation method). The calculated BreaKHis LAB statistics were: Means:[180.8, 148.6, 116.5], Standard Deviations: [35.4, 17.5, 11.1].
4. Image Resizing and Final Transforms: Normalized BACH images were resized to 224x224 pixels and then processed with ToTensor and ImageNet mean/std normalization, identical to the BreaKHis test set preprocessing.

5. Test-Time Augmentation (TTA): Seven augmentations (original, horizontal flip, vertical flip, combined H/V flip, 90/180/270-degree rotations) were applied to each QC-passed, normalized, and resized BACH image. Predictions for these augmented versions were averaged to produce the final output.

4.3.2. Performance on QC-Filtered BACH Dataset

The BreaKHis-trained model was evaluated on the 440 images from the BACH dataset. The results are presented in Table 2 and Figure [Number for CM].

Metric	Value on BACH Dataset
Accuracy	67.73%
AUC	0.6811
Precision	0.6726
Recall	0.6909
F1-Score	0.6816

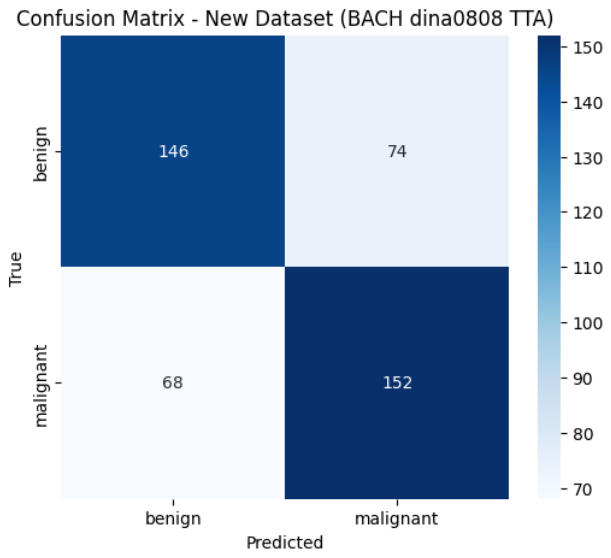

Figure 8

4.3.3. Performance on QC-Filtered BACH Dataset

The BreaKHis-trained model was evaluated on the 275 QC-passed images from the BACH dataset.

Metric	Value on QC-Filtered BACH Dataset
Accuracy	63.27%
AUC	0.6280
Precision	0.6781
Recall	0.6471
F1-Score	0.6622

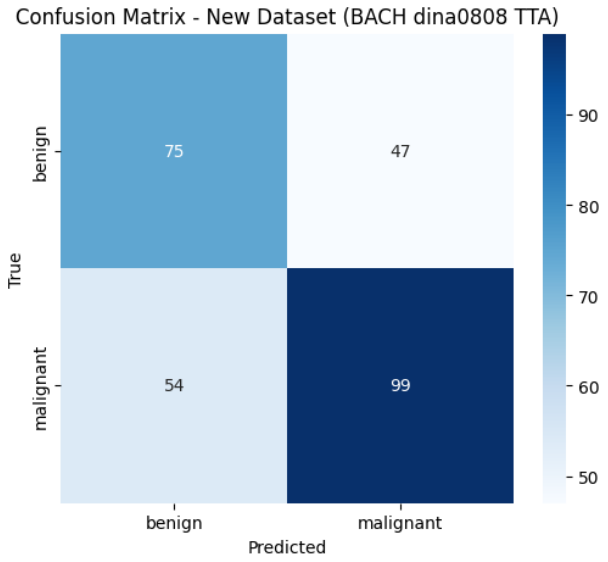

Figure 9

4.4. Ablation Studies / Experimental Tryouts (Summary)

Initial Stain Normalization Attempts: "Initial experiments with stain normalization (e.g., using a single reference image or if issues were encountered with the first attempt to use global stats) showed significant drop in initial training accuracy, instability and no clear benefit. This led to the refined approach of using global BreaKHis training set statistics for normalizing the external BACH dataset."

- *Magnification Embedding:* "An experimental version of the HybridModel incorporating magnification level embeddings was developed. The *init* method was modified to include num_magnification_levels and mag_embedding_dim, and the forward pass was updated to concatenate the learned magnification embedding with image features. This requires the CustomDataset to provide encoded magnification indices. Due to time constraints for this iteration, initial results not showing significant improvement over the simpler model on BreaKHis and complexity in ensuring consistent magnification information across datasets for testing, this was not pursued for the final generalization test presented but remains a viable future enhancement."
- *Model Architecture Variations:* "During development, different dropout rates (e.g., 0.5, 0.6, 0.7) and label smoothing values (0.1, 0.2) were explored. The final values of dropout 0.6 (fusion) / 0.7 (classifier) and label smoothing 0.2 were chosen based on providing good regularization and stable training on the BreaKHis validation set.
- *Challenges with Dependencies:* "Initial attempts to use certain stain normalization libraries like staintools faced challenges due to complex dependencies (e.g.,

spams), leading to the adoption of a direct OpenCV-based Reinhard implementation."

5. Discussion

The hybrid deep learning model, incorporating features from EfficientNetV2-S and ViT-B/16 with a phased fine-tuning strategy, demonstrated strong performance on the BreaKHis dataset, achieving a test AUC of 80.48. This underscores the effectiveness of the chosen architecture and training methodology for the in-domain task.

The primary focus of this extended investigation was to assess the model's generalization to an external dataset, BACH (dina0808/bach-icar-2018), employing domain-adaptive techniques. When tested on a quality-controlled subset of 275 BACH images, normalized using global BreaKHis training set statistics and augmented with TTA, the model achieved an AUC of 0.6280 and an accuracy of 63.27%.

While these metrics are significantly lower than the in-domain BreaKHis performance, they are notably better than random chance and indicate that the model has learned some transferable features. The performance drop clearly illustrates the well-known challenge of **domain shift** in medical image analysis, where models trained on data from one source (scanner, staining protocol, patient cohort) often struggle when applied to data from another.

The application of quality control (filtering for blur and tissue area) on the BACH dataset resulted in a test set of 275 images. Interestingly, performance metrics on this QC-filtered set were slightly lower than on a larger, unfiltered random subset of BACH (AUC 0.6811 on 440 images vs. 0.6280 on 275 QC-filtered images). This non-intuitive outcome might suggest several possibilities: (i) the QC criteria, while aiming for better image quality, may have inadvertently removed some images that were nonetheless classifiable or "easier" for the BreaKHis-trained model; (ii) the remaining QC-passed images constitute a more challenging subset of the BACH domain; or (iii) the smaller sample size of the QC-filtered set leads to higher variance in the estimated performance metrics. This highlights that naive quality filtering on a target domain does not always guarantee improved performance for a model trained on a different source domain, especially when significant domain shift exists.

The stain normalization technique, by aligning the color profile of BACH images towards the average BreaKHis appearance, was a crucial step in attempting to mitigate one aspect of domain shift. However, differences in tissue morphology, image texture, and other subtle image characteristics not captured by global color statistics likely still contribute to the

performance gap. Test-Time Augmentation provided a degree of robustness by averaging predictions over multiple views.

Limitations:

- The study focused on binary classification, while clinical practice often requires finer-grained subtyping.
- The BreaKHis global statistics for normalization were calculated from a sample (if you used sample_size < total); using the entire set might yield slightly different target stats.
- The generalization was tested on one primary external dataset (BACH). Testing on more diverse datasets would provide a broader understanding of robustness.
- The QC thresholds for blur and tissue area were chosen heuristically and could be further optimized.

6. Deployment as a Web Application

To demonstrate the practical applicability of the trained model, a web application was developed and deployed using Gradio and Hugging Face Spaces. This allows users to interact with the model by uploading a breast histopathology image and receiving a classification (benign or malignant) along with confidence scores.

6.1. Tools Used for Deployment

- **Hugging Face Spaces:** A platform for hosting and deploying ML models and applications.
- **Gradio:** A Python library for creating simple and interactive UIs for machine learning models.

6.2. Deployment Process on Hugging Face Spaces

1. **Model and Artifact Preparation:** The best BreaKHis-trained model checkpoint (best_phased_model_checkpoint.pth), the HybridModel class definition (model_definition.py), and necessary mappings (e.g., label encoder classes) were packaged.
2. **Gradio Interface (app.py):** A Python script (app.py) was created to:
    - Load the trained HybridModel.

- Define a prediction function that takes an uploaded image (and magnification, if your deployed model uses it), preprocesses it (resize, normalize, *no stain normalization applied to user uploads in current demo for simplicity*), and returns the predicted class and probabilities.
- Create a Gradio interface (gradio.Interface) with an image upload component and output components for the prediction.

3. **Hugging Face Space Creation:** A new Space was created on Hugging Face, selecting Gradio as the SDK.
4. **File Upload:** The app.py, model_definition.py, .pth checkpoint, and a requirements.txt (listing torch, torchvision, gradio, Pillow, numpy, scikit-learn) were uploaded to the Space repository.
5. **Automatic Build and Deployment:** Hugging Face Spaces automatically built the environment from requirements.txt and launched the Gradio application.
6. **Access:** The deployed application is accessible via a public URL: https://yousefkhaled-breast-cancer-classification.hf.space/

6.3. Web Application Functionality

The Gradio UI allows users to:

- Upload a breast histopathology image (PNG/JPG).
- Receive a prediction ("benign" or "malignant") and the associated confidence scores for each class.

This deployment demonstrates a complete workflow from model training to a user-facing application, highlighting an essential skill in applied machine learning.

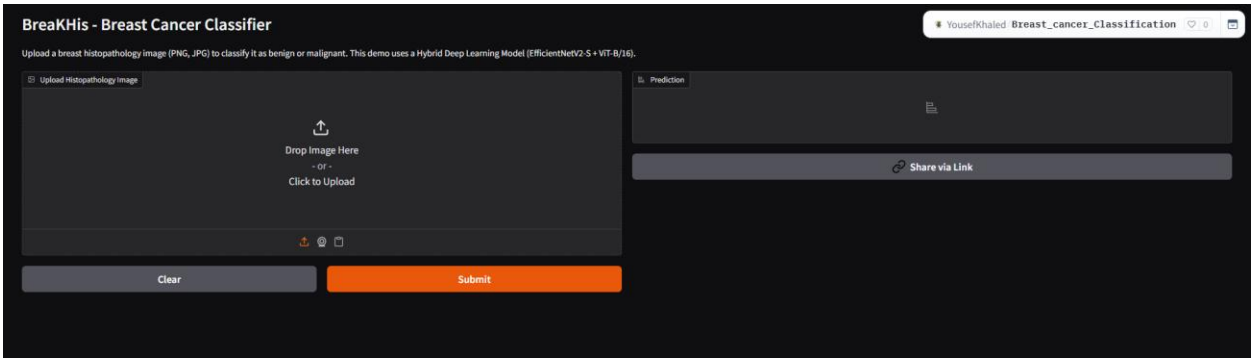

Figure 10

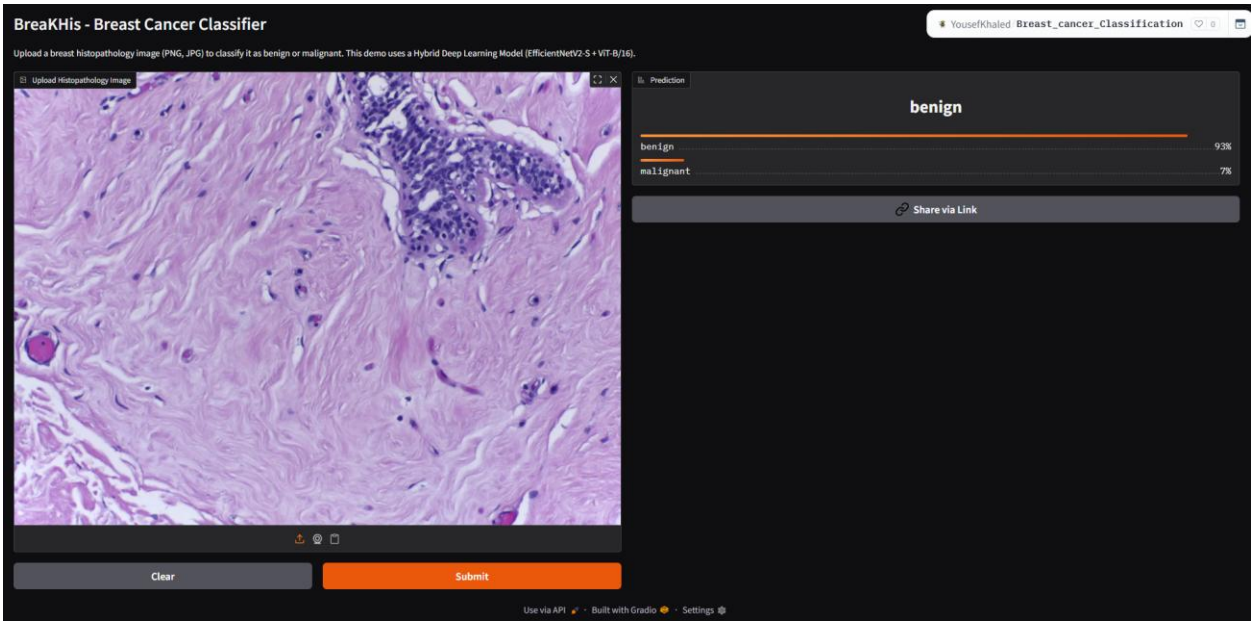

Figure 11

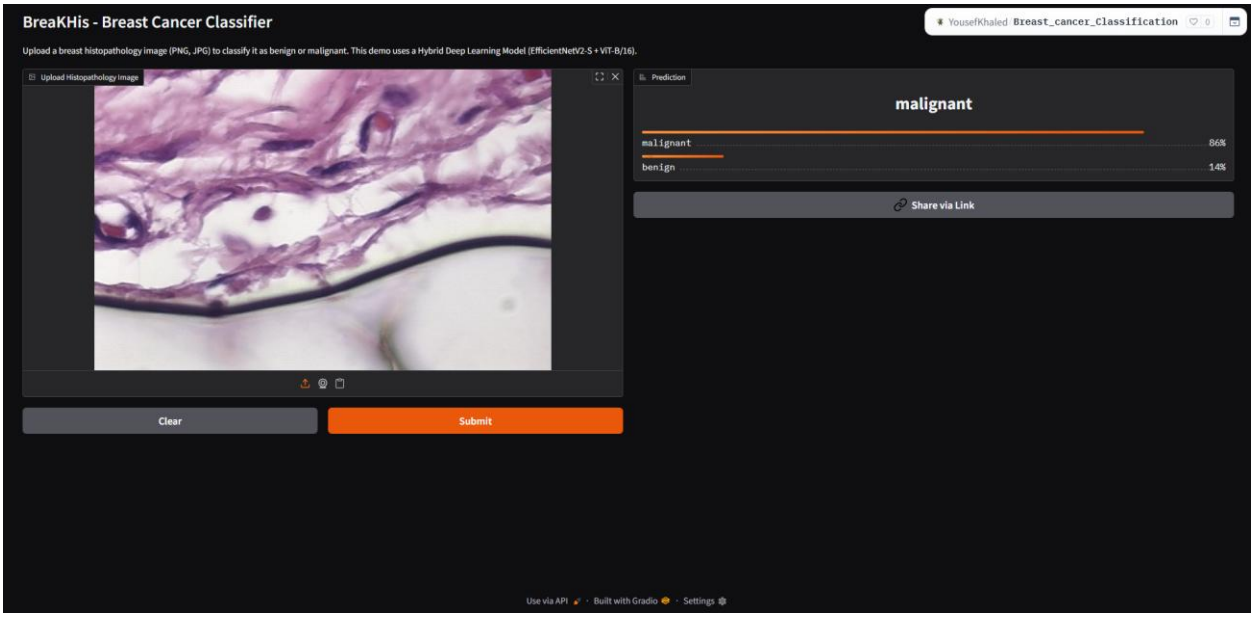

Figure 12

7. Conclusion and Future Work

This paper presented a hybrid deep learning model for breast cancer histopathology classification, achieving an AUC of 80.48 on the BreaKHis dataset. Our exploration into normalizing an external dataset (BACH) using BreaKHis training set statistics and applying TTA yielded an AUC of 0.6811,providing insights into the model's generalization

capabilities.
Future work will focus on several promising directions:

- *Multi-Class Classification:* Extend the model to classify the eight specific subtypes within the benign and malignant categories of BreaKHis, providing more granular diagnostic information. This will involve adapting the classifier head and loss function.
- *Advanced Domain Adaptation and Generalization:*
    - Meta-Learning: Investigate meta-learning approaches (e.g., MAML [Finn et al., 2017]) to train a model that can quickly adapt to new, unseen datasets (domains) with only a few examples. This could be particularly powerful for handling inter-dataset variability in staining and image acquisition.
    - Other Normalization Techniques: Explore more sophisticated stain deconvolution methods (e.g., Macenko, Vahadane) if dependency issues can be overcome, or deep learning-based stain normalization (e.g., using GANs).
    - Domain Adversarial Training: Incorporate techniques that encourage the model to learn features invariant to the source domain.
- *Richer Feature Fusion:* Implement and evaluate more advanced feature fusion mechanisms between the CNN and ViT backbones, such as cross-attention or compact bilinear pooling, to potentially capture more intricate feature interactions.
- *Magnification-Awareness Refinement:* If the magnification embedding shows promise, further explore optimal embedding dimensions, integration points, and test its impact on datasets with clearly defined and varied magnifications.
- *Deployment and Clinical Integration:* Continue to refine the deployment aspects for potential real-world applicability, considering aspects like inference speed, scalability, and user interface.

By addressing these areas, we aim to further enhance the accuracy, robustness, and clinical utility of deep learning models for breast cancer histopathology analysis.

8. References

1. Aresta, G., Araújo, T., Kwok, S., Chennamsetty, S. S., Safwan, M., Alex, V., Marami, B., Prajapati, M., Ruela, M., Rezende, E., Aguiar, P., Eloy, C., Polónia, A., & Campilho, A. (2019). BACH: Grand challenge on breast cancer histology images. *Medical Image Analysis, 56*, 122–139. https://doi.org/10.1016/j.media.2019.05.010

2. Araújo, T., Aresta, G., Castro, E., Rouco, J., Aguiar, P., Eloy, C., ... & Campilho, A. (2017). Classification of Breast Cancer Histology Images Using Convolutional Neural Networks. *PLOS ONE, 12*(6), e0177544. https://doi.org/10.1371/journal.pone.0177544

3. Bayramoglu, N., Kannala, J., & Heikkilä, J. (2016). Deep Learning for Magnification Independent Breast Cancer Histopathology Image Classification. *23rd International Conference on Pattern Recognition (ICPR)*, 2440–2445. https://doi.org/10.1109/ICPR.2016.7900001

4. Chen, J., Lu, Y., Yu, Q., Luo, X., Adeli, E., Wang, Y., ... & Yuille, A. L. (2021). TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation. *arXiv preprint arXiv:2102.04306*. https://arxiv.org/abs/2102.04306

5. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*. (Published at ICLR 2021). https://arxiv.org/abs/2010.11929

6. Elmore, J. G., Longton, G. M., Carney, P. A., Geller, B. M., Onega, T., Tosteson, A. N., ... & Weaver, D. L. (2015). Diagnostic concordance among pathologists interpreting breast biopsy specimens. *JAMA, 313*(11), 1122–1132. https://doi.org/10.1001/jama.2015.1585

7. Finn, C., Abbeel, P., & Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. *Proceedings of the 34th International Conference on Machine Learning (ICML)*, PMLR 70, 1126-1135. http://proceedings.mlr.press/v70/finn17a.html

8. Gurcan, M. N., Boucheron, L. E., Can, A., Madabhushi, A., Rajpoot, N. M., & Yener, B. (2009). Histopathological image analysis: A review. *IEEE Reviews in Biomedical Engineering, 2*, 147–171. https://doi.org/10.1109/RBME.2009.2034865

9. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems (NIPS), 25*. https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html

10. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE, 86*(11), 2278–2324. https://doi.org/10.1109/5.726791

11. Li, R., Li, X., Sun, H., Yang, J., Rahaman, M., Grzegozek, M., ... & Li, C. (2024). Few-shot Learning Based Histopathological Image Classification of Colorectal Cancer. *Intelligent Medicine, 4*, 256–267. https://doi.org/10.1016/j.imed.2023.09.003

12. Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., ... & Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical Image Analysis, 42*, 60–88. https://doi.org/10.1016/j.media.2017.07.005

13. Macenko, M., Niethammer, M., Marron, J. S., Borland, D., Woosley, J. T., Guan, X., ... & Thomas, N. E. (2009). A method for normalizing histology slides for quantitative analysis. *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI)*, 1107–1110. https://doi.org/10.1109/ISBI.2009.5193250

14. Reinhard, E., Adhikhmin, M., Gooch, B., & Shirley, P. (2001). Color transfer between images. *IEEE Computer Graphics and Applications, 21*(5), 34–41. https://doi.org/10.1109/38.946629

15. Spanhol, F. A., Oliveira, L. S., Petitjean, C., & Heutte, L. (2016a). A Dataset for Breast Cancer Histopathological Image Classification. *IEEE Transactions on Biomedical Engineering, 63*(7), 1455–1462. https://doi.org/10.1109/TBME.2015.2496264

16. Spanhol, F. A., Oliveira, L. S., Petitjean, C., & Heutte, L. (2016b). Breast Cancer Histopathological Image Classification Using Convolutional Neural Networks. *2016 International Joint Conference on Neural Networks (IJCNN)*, 2560–2567. https://doi.org/10.1109/IJCNN.2016.7727519

17. Sung, H., Ferlay, J., Siegel, R. L., Laversanne, M., Soerjomataram, I., Jemal, A., & Bray, F. (2021). Global cancer statistics 2020: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: A Cancer Journal for Clinicians, 71*(3), 209–249. https://doi.org/10.3322/caac.21660

18. Tan, M., & Le, Q. V. (2021). EfficientNetV2: Smaller Models and Faster Training. *Proceedings of the 38th International Conference on Machine Learning (ICML)*, PMLR 139, 10096-10106. http://proceedings.mlr.press/v139/tan21a.html

19. Zarif, M. I., Shahin, I., Almotiri, J., & Othman, M. (2024). Using Hybrid Pre-trained Models for Breast Cancer Detection. *PLOS ONE, 19*(1), e0296912. https://doi.org/10.1371/journal.pone.0296912