# Breast Cancer Classification Using Deep Learning: A DenseNet121 and Grad-CAM Approach

**By ARAFAT SANI**

*An exploration into building an interpretable AI to assist in histopathological diagnosis, focusing on methodology, performance, and the challenges of fine-grained classification.*

---

## Abstract

Breast cancer remains a formidable global health challenge and is the leading cause of cancer-related mortality among women. The clinical gold standard for diagnosis relies on the histopathological examination of tissue slides, a meticulous but time-consuming and subjective process. This report details the development of an end-to-end deep learning pipeline designed to classify breast histopathology images from the BreaKHis dataset. At its core, the system employs a DenseNet121 architecture, structured hierarchically to first perform a binary classification between benign and malignant tissues. Following this, dedicated models attempt a more granular classification into four benign and four malignant subtypes. A critical component of this research is the integration of Gradient-weighted Class Activation Mapping (Grad-CAM), a technique used to demystify the model's decisions by visualizing the image

regions that most influence its predictions. This addresses the "black box" problem often associated with neural networks. Our binary model achieved a promising accuracy of approximately **84%** on the test set. However, the subtype classification models encountered significant challenges, with performance dropping considerably, especially for malignant subtypes. The Grad-CAM visualizations revealed that correct predictions often correlated with clinically relevant histologic structures, whereas failures were linked to diffuse or mislocalized model attention. This paper provides a comprehensive overview of the data preprocessing, model design, training strategy, and evaluation, while also candidly discussing the project's limitations and outlining a roadmap for future improvements.

## 1. Introduction: The Challenge and Promise of AI in Pathology

The diagnosis of breast cancer is a cornerstone of modern oncology, traditionally resting in the hands of skilled pathologists. Their work involves a detailed visual inspection of tissue samples stained with hematoxylin and eosin (H&E), where they search for tell-tale architectural and cellular features that distinguish healthy tissue from benign growths and malignant cancers. While this method has been refined over decades and remains highly effective, it is not without its challenges. The process is labor-intensive, requiring pathologists to scrutinize numerous high-resolution images, and is susceptible to inter-observer variability—where two experts might interpret the same subtle or borderline case differently.

This is where artificial intelligence, specifically deep learning and Convolutional Neural Networks (CNNs), offers a transformative potential. CNNs are designed to automatically learn discriminative features directly from image data, mimicking the human visual cortex. They have demonstrated remarkable success in various medical imaging domains, promising a more systematic, quantitative, and consistent approach to analysis. This project's objective is not to replace the pathologist but to create an **interpretable classification system** that can serve as a powerful assistive tool, augmenting the diagnostic process by flagging areas of interest and providing a reliable "second opinion."

To achieve this, we selected the **DenseNet121** architecture. Its unique "dense connectivity" pattern, where each layer is connected to every other layer in a feed-forward fashion, encourages efficient feature reuse and improves the flow of gradients during training. This makes it particularly well-suited for medical imaging tasks where datasets may not be massive. Crucially, to build trust and facilitate clinical validation, we explicitly integrated **Grad-CAM**. This technique generates a localization heatmap for every prediction, providing a transparent view into which regions of the tissue image the model found most important for its decision. This explainability is paramount for any AI tool intended for clinical use.

## 2. The BreaKHis Dataset: Our Digital Microscope

The foundation of any machine learning project is its data. For this study, we utilized the publicly available **BreaKHis dataset**, a comprehensive collection of 7,909 microscopic images of breast tumor tissue. These images are stained with H&E and were captured at four different magnification levels (40×, 100×, 200×, and 400×), providing a rich source of visual information at varying scales.

Each image in the dataset is annotated with a two-level hierarchical label, which is essential for our multi-stage classification approach:

- **Binary Label:** The primary distinction is between *Benign* (non-cancerous) and *Malignant* (cancerous) tissue.
- **Subtype Label:** Within each binary category, a more specific histological subtype is provided.
  - **Benign Subtypes:** Adenosis (A), Fibroadenoma (F), Phyllodes Tumor (P), and Tubular Adenoma (T).
  - **Malignant Subtypes:** Ductal Carcinoma (D), Lobular Carcinoma (L), Mucinous Carcinoma (M), and Papillary Carcinoma (P).

A key aspect of working with this dataset involved programmatically extracting these labels from the image filenames, which follow a consistent naming convention. To ensure the robustness and generalizability of our models, the data was carefully

partitioned into training, validation, and test sets. We employed **stratified sampling** based on the binary label. This technique ensures that the proportion of benign and malignant cases is consistent across all three sets, preventing a scenario where the model is trained on one distribution of data and tested on another.

## 3. Methodology: Building the Classifier

Our methodology was designed to be systematic, reproducible, and transparent. It encompasses the entire pipeline from setting up the environment to implementing the final explainability layer.

### 3.1 Environment and Reproducibility

All experiments were conducted within a Kaggle environment, leveraging the power of TensorFlow and Keras for model development. To ensure that our results are scientifically valid and can be reproduced by other researchers, we fixed the random seeds for Python, NumPy, and TensorFlow. This guarantees that any process with a random element—such as weight initialization or data shuffling—produces the same outcome every time the code is run.

### 3.2 Preprocessing and Data Augmentation

Raw images from the dataset first underwent a series of preprocessing steps to prepare them for the model. Each PNG image was read, decoded into an RGB tensor, and resized to a uniform 128×128 pixels. This standardization is crucial for feeding data into the neural network. Normalization was then applied; if using pretrained ImageNet weights, we used the specific normalization function for DenseNet, otherwise, pixel values were scaled to a [0, 1] range.

To combat overfitting and help the model generalize better to new, unseen images, we applied **data augmentation** during training. This involves creating modified versions of the training images on-the-fly. Our augmentation strategy included:

- Random horizontal flips

- Small random rotations and zooms
- Minor contrast adjustments

These transformations mimic realistic variations in specimen preparation and imaging without altering the underlying diagnostic features, effectively expanding the training set and making the model more robust.

### 3.3 Dataset Construction with `tf.data`

For maximum efficiency, we constructed three distinct data pipelines using TensorFlow's `tf.data` API: one for the binary task (benign vs. malignant), one for benign subtype classification, and one for malignant subtype classification. This API is highly optimized for performance, allowing us to map file paths to preprocessed images, apply augmentations, shuffle the data with a large buffer, group images into batches for GPU processing, and prefetch data to ensure the GPU never sits idle waiting for the CPU.

### 3.4 Model Architecture: DenseNet121 at the Core

The heart of our classifier is the **DenseNet121** model, used as a feature extraction "backbone." We used the version pre-trained on ImageNet but removed its final classification layer (`include_top=False`). On top of this powerful backbone, we attached a new, compact classification "head" tailored to our specific tasks. This head consists of:

1. **Global Average Pooling:** To condense the feature maps from the backbone into a single feature vector per image.
2. **A Fully Connected Layer:** A dense layer with 64 units and ReLU activation, with L2 regularization to prevent overfitting.
3. **Batch Normalization:** To stabilize and accelerate training.
4. **Dropout:** A regularization technique where 40% of neurons are randomly dropped during training to prevent co-adaptation.
5. **A Final Softmax Layer:** This output layer has a size matching the number of classes (2 for the binary task, 4 for each subtype task) and produces the final

class probabilities.

Initially, the backbone layers were "frozen," meaning only the weights of our new head were trained. This allows the head to learn from the rich features provided by the pre-trained backbone without destabilizing them.

### 3.5 The Training Strategy: A Three-Model Approach

We trained three separate models, each specialized for its task. All models were trained using the **Adam optimizer** with a learning rate of 1e-4, minimizing the categorical cross-entropy loss function. To ensure optimal performance and prevent overfitting, we employed two key callbacks:

- **Early Stopping:** This monitored the validation loss and would halt training if no improvement was seen after a set number of epochs, restoring the model weights from the best-performing epoch.
- **Learning Rate Scheduler:** If the validation loss plateaued, this callback would reduce the learning rate, allowing the model to make finer adjustments and find a better minimum.

### 3.6 Explainability: Peeking Inside the Black Box with Grad-CAM

A core tenet of this project was to create an auditable system. To this end, we implemented **Gradient-weighted Class Activation Mapping (Grad-CAM)**. This technique provides visual explanations for the model's predictions. For any given image, Grad-CAM computes the gradient of the predicted class score with respect to the feature maps of the final convolutional layer. These gradients are used to produce a heatmap that highlights the regions of the image that were most influential in the model's decision. By overlaying this heatmap on the original image, we can directly visualize the model's "attention," making its behavior transparent and interpretable for both error analysis and communication with clinical stakeholders.

## 4. Results: Performance and Pitfalls

---

The evaluation of our three models on the held-out test set revealed a stark contrast in performance between the high-level binary task and the more granular subtype classification tasks. This section details the quantitative results and provides an analysis of the models' behaviors.

## 4.1 Binary Classification (Benign vs. Malignant): A Clear Success

The binary model, tasked with the most clinically critical distinction, performed admirably, achieving a test accuracy of approximately **~84%**. This result indicates that the DenseNet121 backbone, even with a limited input resolution of 128x128, is capable of learning robust features to reliably separate benign from malignant tissue. Analysis of the confusion matrix showed a strong ability to correctly identify malignant cases, which is crucial. There was some modest confusion where benign images were misclassified as malignant, suggesting a potentially conservative bias. In a clinical setting, a higher false positive rate is often more acceptable than a higher false negative rate (missing a cancer).

## 4.2 Benign Subtype Classification: A Significant Challenge

The benign subtype model's performance dropped significantly, attaining a test accuracy of only **~44%**. While this is better than random chance (25%), it reveals substantial confusion between the four benign subtypes. The confusion matrix showed that the model was heavily biased towards predicting **Fibroadenoma (F)**, regardless of the true label. This could stem from several factors: a class imbalance in the training data, the visual dominance of F-like textures, or the 128x128 resolution being insufficient to capture the subtle histological differences that separate these benign conditions.

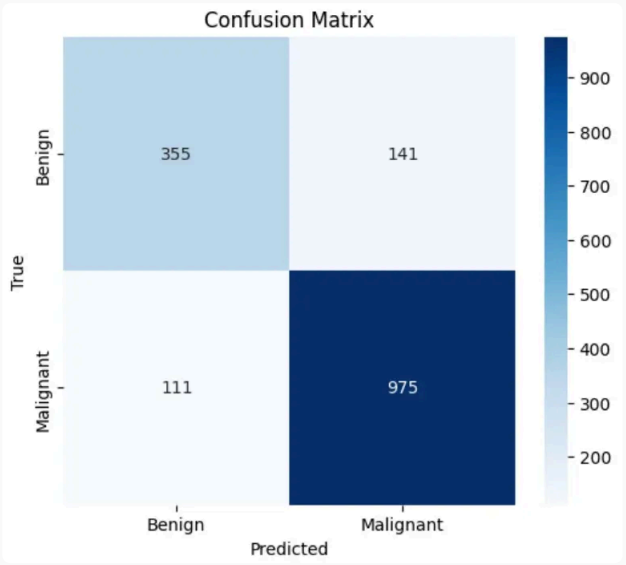
## 4.3 Malignant Subtype Classification: The Toughest Task

The challenge intensified with the malignant subtype model, which yielded a test accuracy of a mere **~15%**. This underscores the extreme difficulty of discriminating between malignant histologies using small image patches and a frozen backbone. The confusion matrix revealed a profound bias, with the model heavily assigning

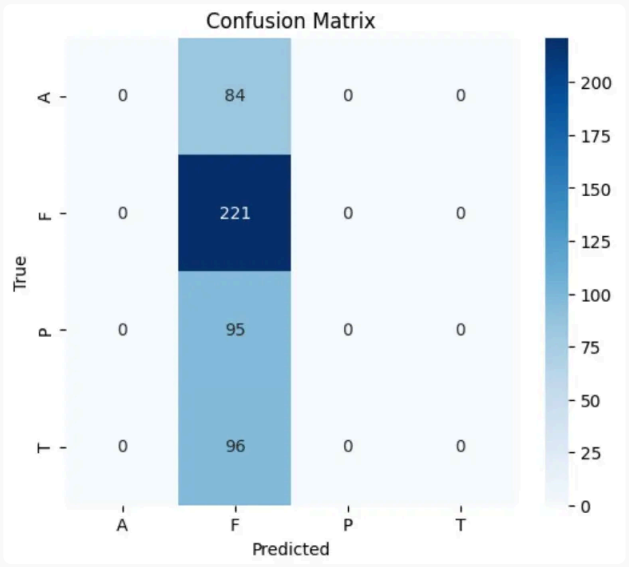
predictions to **Mucinous Carcinoma (M)**. This suggests the model latched onto certain textural patterns associated with 'M' and failed to learn the more nuanced features of other subtypes. This limitation highlights the need for more advanced techniques, such as higher resolution inputs or targeted fine-tuning of the model.

## 4.4 Visualizing Performance: The Confusion Matrices

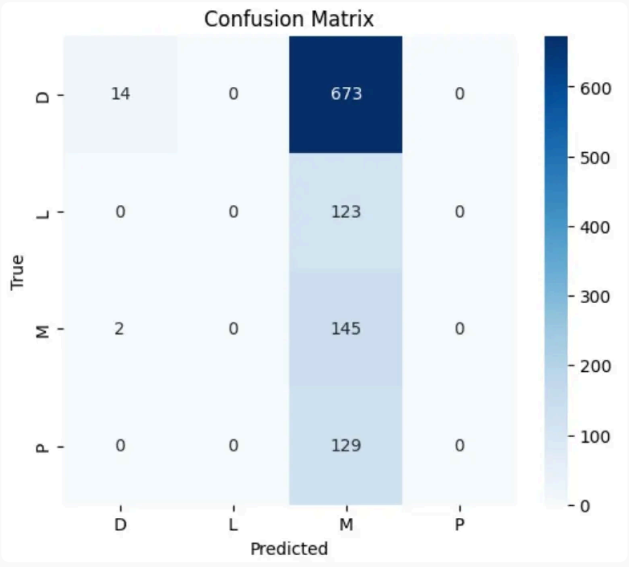
To delve deeper into the classification behavior, we generated confusion matrices for each model. These matrices provide a clear visual breakdown of correct and incorrect predictions for each class.




***Binary Classification Model:*** *The matrix shows strong performance.* ***975 malignant samples*** *were correctly classified (True Positives) and* ***355 benign samples*** *were correctly classified (True Negatives). The model misclassified 141 benign samples as malignant (False Positives) and 111 malignant samples as benign (False Negatives). The low false negative rate is a particularly encouraging result for potential clinical applications.*




***Benign Subtype Model:*** *This matrix clearly illustrates the model's primary weakness. A large number of predictions are concentrated in the 'F' (Fibroadenoma) column, regardless of the true class (A, F, P, or T). This indicates a failure to differentiate between the benign subtypes, likely due to feature overlap or data imbalance.*



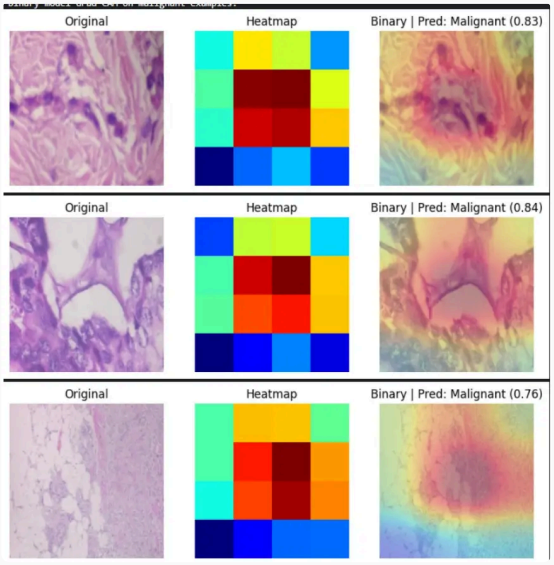

***Malignant Subtype Model:*** *Similar to the benign model, this matrix reveals a strong prediction bias, this time towards the 'M' (Mucinous Carcinoma) subtype. For instance, 673 Ductal Carcinoma (D) cases were misclassified as 'M'. This highlights a significant challenge in learning discriminative features for these visually similar and complex malignant subtypes.*

## 5. Interpretability in Action: Visualizing Decisions with Grad-CAM

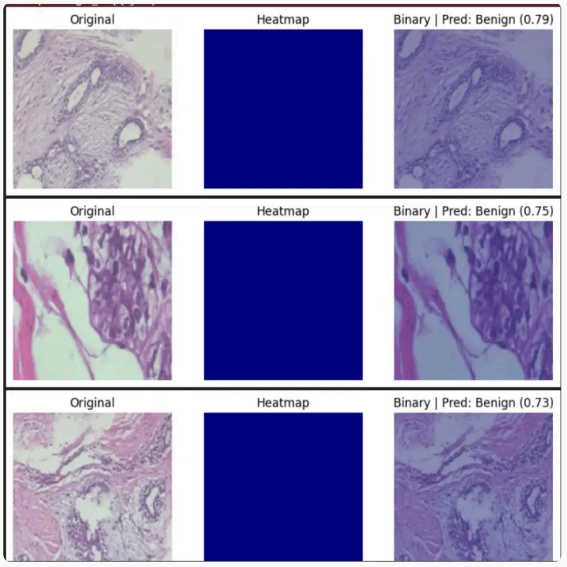
Beyond raw accuracy scores, understanding *why* a model makes a certain prediction is crucial. The Grad-CAM visualizations provide this critical insight, acting as a window into the model's decision-making process. We used this technique to both validate correct predictions and diagnose failure modes.

## 5.1 Quick Grad-CAM Examples

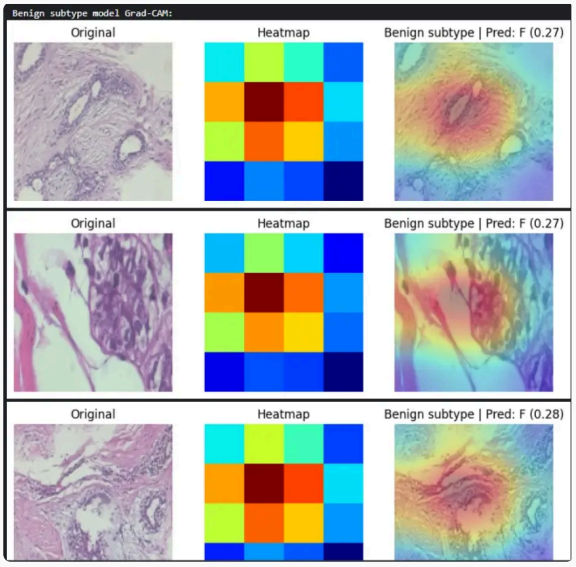
Initial spot-checks confirmed that the models were often attending to plausible histologic structures.



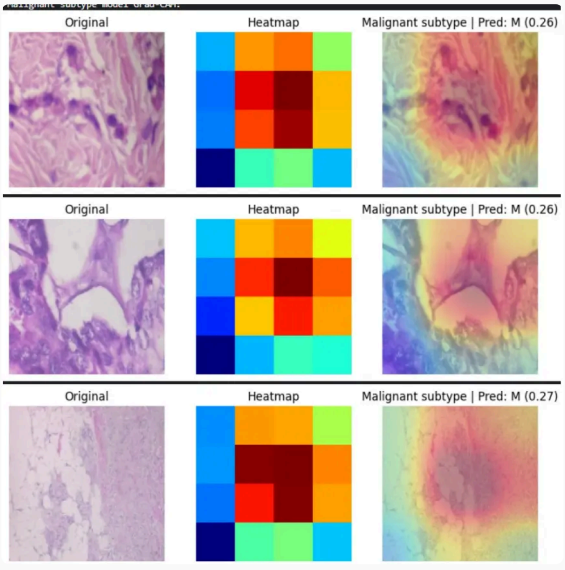

***Binary Model (Malignant):*** *In correctly identified malignant cases, the model's attention (red heatmap) often concentrates on dense cellular clusters and irregular glandular structures—key indicators used by pathologists.*



***Binary Model (Benign):*** *For benign examples, the heatmaps are typically weaker and more diffuse. This is consistent with the absence of overt malignant features, with attention still localizing near normal stromal and ductal regions.*



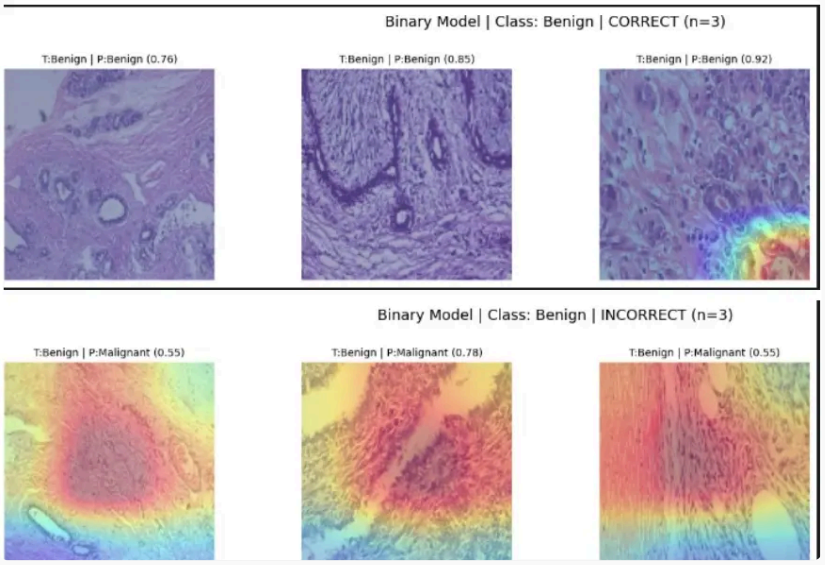

***Benign Subtype Model:*** *The overlays for the benign subtype model often highlight glandular lumens and surrounding stroma. However, the patterns can be similar across different subtypes, helping to explain the confusion towards the 'F' class seen in the results.*



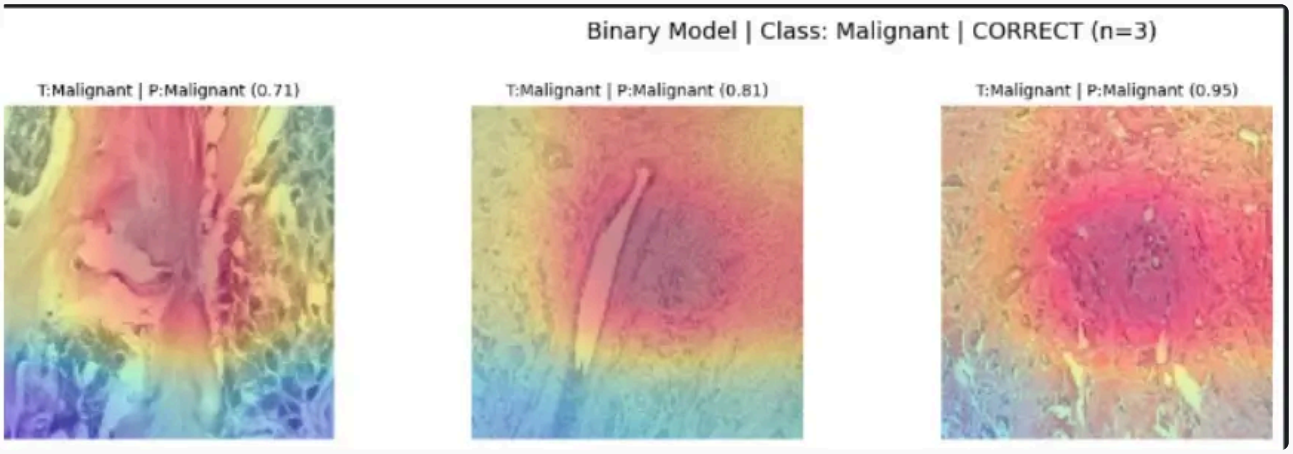

***Malignant Subtype Model:*** *The attention maps for the malignant subtype model are often diffuse or misaligned, which correlates directly with its low classification accuracy. This suggests the model struggled to find consistent, discriminative features for this task.*

## 5.2 Systematic Analysis: Correct vs. Incorrect Predictions

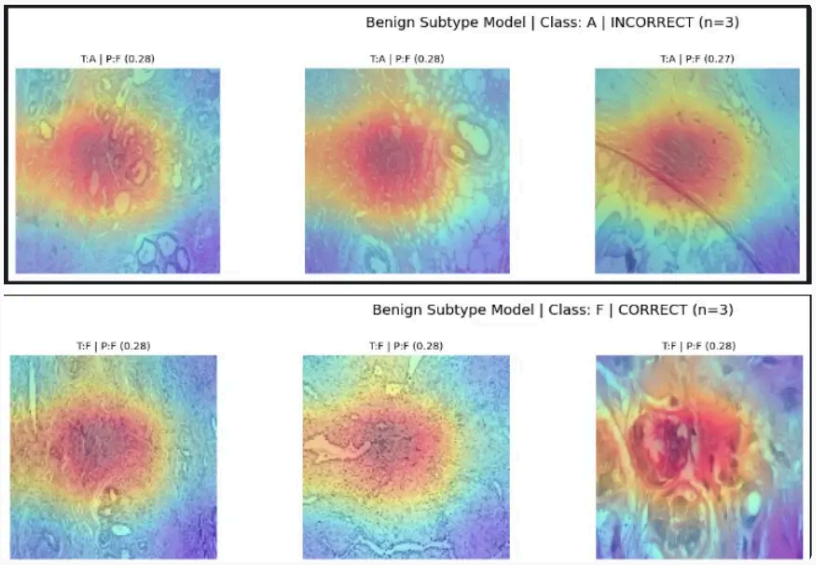
To systematically assess model behavior, we generated panels showing Grad-CAM overlays for multiple correct and incorrect predictions for each class. These grids are invaluable for identifying class-specific failure modes.



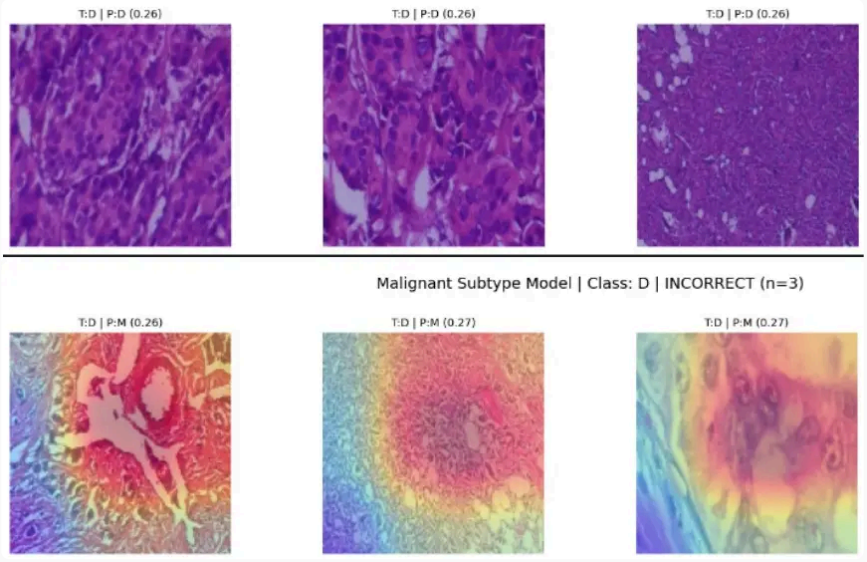

***Binary Model - Malignant Class (Correct):*** *This panel reinforces the model's trustworthiness. For correctly identified malignant images, the heatmaps consistently show focused attention on suspicious, high-density cell structures.*



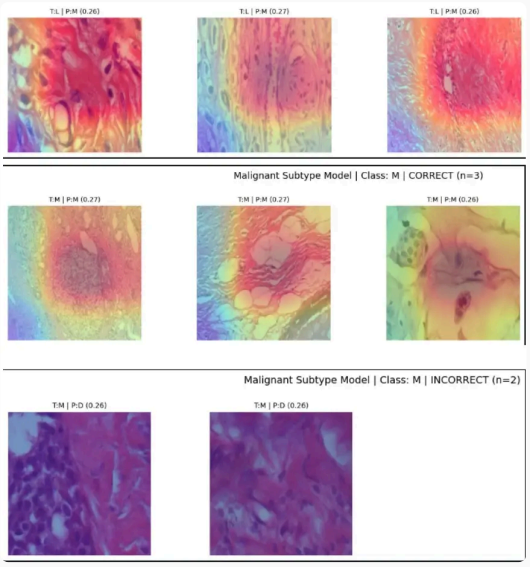

***Binary Model - Malignant Class (Incorrect):*** *In cases where a malignant image was misclassified as benign, the failure mode is clear. The model's attention is either misplaced on background areas or is too diffuse to capture the key malignant focus.*




***Binary Model - Benign Class (Correct and Incorrect):*** *Correct benign predictions (top row) show moderate, localized attention. Incorrect predictions (bottom row), where benign is misclassified as malignant, often display stronger, broader heatmaps, suggesting the model is over-interpreting certain features as suspicious.*




***Malignant Subtype Model (Correct):*** *Even in the few correct predictions, the attention can be varied, highlighting the difficulty of the task. The model appears to be focusing on different textural cues for each subtype.*




***Malignant Subtype Model (Incorrect):*** *The incorrect predictions for malignant subtypes are often characterized by diffuse heat or attention focused on non-diagnostic background areas, confirming that the model failed to locate the relevant distinguishing features.*

## 6. Discussion: Synthesizing the Findings

The experimental results paint a clear picture: our deep learning pipeline, centered on DenseNet121, is highly effective for the primary binary classification of breast cancer histopathology images. The **~84%** test accuracy, combined with Grad-CAM visualizations showing attention on morphologically relevant regions, demonstrates the model's potential as a reliable diagnostic aid. This success holds even when the model is trained from scratch, highlighting the power of the DenseNet architecture.

However, the project also starkly illuminates the challenges of fine-grained classification. The significant drop in performance for the subtype tasks, particularly for malignant categories, is a critical finding. This performance gap likely arises from a confluence of factors:

- **Limited Data Per Subtype:** While the overall dataset is large, the number of examples for each of the eight subtypes is much smaller, making it difficult for the model to learn robust representations for each one.
- **Subtle Inter-Class Variation:** The visual differences between some subtypes can be extremely subtle, requiring a level of detail that may be lost at our chosen input

resolution.

- **Low Input Resolution:** The 128x128 pixel input size, chosen for computational efficiency, likely constrains the model's ability to capture the fine-grained cytologic details necessary for accurate subtype differentiation.

The role of **Grad-CAM** in this project cannot be overstated. It transitioned from being a simple visualization tool to a core component of our analysis and debugging process. It provided compelling evidence that correct predictions were often based on sound "reasoning" (i.e., focusing on relevant pathology), while failures could be directly attributed to diffuse or mislocalized attention. The "Auto Panels" were particularly insightful, revealing systemic issues like the model's attraction to Fibroadenoma ('F') and Mucinous ('M') textures, which points towards specific avenues for improvement, such as data rebalancing or the use of class-aware loss functions.

From a real-world perspective, the implications are twofold. The binary model shows significant promise and could be developed further into a screening or second-opinion tool to help pathologists prioritize cases and reduce workload. The subtype models, in their current form, are not clinically viable but serve as a crucial baseline, highlighting the specific technical hurdles that must be overcome to achieve automated fine-grained cancer classification.

## 7. Conclusion and Future Work: The Path Forward

This project successfully delivered a complete, interpretable pipeline for histopathology image analysis using DenseNet121 and Grad-CAM on the BreaKHis dataset. We achieved strong performance on the clinically vital binary task of distinguishing benign from malignant tissue and established an important baseline for the more challenging task of subtype recognition. The work underscores both the potential of deep learning in pathology and the critical importance of model interpretability.

The challenges encountered pave a clear path for future work. To build upon this foundation, we propose the following key improvements:

1. **Increase Image Resolution:** Moving to higher resolutions (e.g., 224×224 or 512x512) is a top priority to provide the model with more fine-grained detail.
2. **Fine-Tune the Backbone:** Unfreezing the later blocks of the DenseNet backbone and fine-tuning them on our specific dataset could allow the model to adapt its feature extractors more closely to histopathology images.
3. **Address Class Imbalance:** Implement advanced strategies like class-based over-sampling (e.g., SMOTE) or using cost-sensitive loss functions that penalize misclassifications of minority classes more heavily.
4. **Explore Multi-Task Learning:** Train a single model to jointly predict both the binary and subtype labels. This could allow the model to learn more robust, shared representations.
5. **Investigate Advanced Architectures:** Explore newer architectures, such as Vision Transformers (ViTs), which may be better at integrating local details with broader tissue context.
6. **Magnification-Specific Analysis:** Analyze performance by magnification level and potentially train expert models specialized for each magnification.

## 8. Reproducibility Notes

The project was designed with reproducibility in mind. The code is implemented using efficient `tf.data` pipelines and supports offline execution by allowing the DenseNet121 model to be initialized from scratch without downloading ImageNet weights. The use of fixed random seeds across all relevant libraries ensures that the data splits, model initializations, and training processes are deterministic. Training is governed by automated early stopping and a learning-rate scheduler to ensure stable and consistent convergence. All figures and analysis panels were generated programmatically using shared utility functions.

## 9. Appendix: Code Snippets

### Code for Quick Grad-CAM Figures

```
binary_names    = ['Benign', 'Malignant']
benign_names    = ['A','F','P','T']
malignant_names = ['D','L','M','P']

# Sample a few images per binary class (if available)
benign_paths = test_df[test_df['binary_label']==0]['filepath'] \
    .sample(min(3, (test_df['binary_label']==0).sum()), random_state=SEED).
malig_paths  = test_df[test_df['binary_label']==1]['filepath'] \
    .sample(min(3, (test_df['binary_label']==1).sum()), random_state=SEED).

print("Binary model Grad-CAM on benign examples:")
for p in benign_paths:
    show_gradcam(binary_model, p, binary_names, title_prefix='Binary | ')

print("Binary model Grad-CAM on malignant examples:")
for p in malig_paths:
    show_gradcam(binary_model, p, binary_names, title_prefix='Binary | ')

print("Benign subtype model Grad-CAM:")
for p in benign_paths:
    show_gradcam(benign_model, p, benign_names, title_prefix='Benign subtyp

print("Malignant subtype model Grad-CAM:")
for p in malig_paths:
    show_gradcam(malig_model, p, malignant_names, title_prefix='Malignant s
```

## Code for Auto Panels (Correct/Incorrect Grids)

```
def predict_on_paths(model, paths, batch_size=32):
    """Run model.predict on raw file paths with the same preprocessing used
    def gen():
        for p in paths:
            imbytes = tf.io.read_file(p)
            img = tf.image.decode_png(imbytes, channels=3)
            img = tf.image.resize(img, IMG_SIZE)
            img = tf.cast(img, tf.float32)
            img = preprocess_fn(img)
            yield img
    ds = tf.data.Dataset.from_generator(
        gen,
        output_signature=tf.TensorSpec(shape=(IMG_SIZE[0], IMG_SIZE[1], 3),
```

```
    ).batch(batch_size)
    return model.predict(ds, verbose=0)

def build_predictions_df(model, df_subset, label_col, class_names):
    df_local = df_subset.dropna(subset=[label_col]).copy()
    paths = df_local['filepath'].tolist()
    y_true = df_local[label_col].astype(int).to_numpy()
    preds = predict_on_paths(model, paths)
    y_pred = np.argmax(preds, axis=1)
    y_prob = np.max(preds, axis=1)
    out = pd.DataFrame({'filepath': paths, 'true': y_true, 'pred': y_pred,
    out['correct'] = (out['true'] == out['pred']).astype(int)
    out['true_name'] = [class_names[i] for i in out['true']]
    out['pred_name'] = [class_names[i] for i in out['pred']]
    return out

def sample_per_class(df_pred, N=3):
    """Return dict[class_id] = (correct_df, incorrect_df) with up to N samp
    result = {}
    for c in sorted(df_pred['true'].unique()):
        df_c = df_pred[df_pred['true'] == c]
        corr = df_c[df_c['correct'] == 1].sample(min(N, (df_c['correct']==1
        inc  = df_c[df_c['correct'] == 0].sample(min(N, (df_c['correct']==0
        result[c] = (corr, inc)
    return result

def plot_gradcam_grid(model, df_rows, class_names, title, cols=4, last_conv
    # ... (Implementation for plotting grid) ...

def render_panels_for_model(model, df_pred, class_names, panel_title_prefix
    # ... (Implementation for rendering panels) ...

# Run panels
N_SAMPLES = 3
GRID_COLS = 4

# Binary panels
binary_names = ['Benign', 'Malignant']
# ... (Code to build df and render panels) ...

# Benign subtype panels
benign_names = ['A','F','P','T']
# ... (Code to build df and render panels) ...

# Malignant subtype panels
malignant_names = ['D','L','M','P']
# ... (Code to build df and render panels) ...
```

## 10. Annotated Bibliography


1. **[1] M. A. Araújo, G. Aresta, E. P. Silva, P. Aguiar, C. Eloy, and A. Campilho, "Classification of breast cancer histology images using Convolutional Neural Networks," *PLoS One*, vol. 12, no. 6, p. e0177544, 2017.**

   > *This study was a primary framing resource for our project. It applies CNNs to the same BreaKHis dataset for both binary and multi-class classification, providing essential baseline performance metrics and informing our preprocessing decisions.*

2. **[2] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4700–4708.**

   > *This is the foundational paper that introduces the DenseNet architecture. It justifies our choice of backbone, explaining how its unique connectivity pattern improves gradient flow and feature reuse, which is particularly beneficial for achieving robust learning on limited datasets.*

3. **[3] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 618–626.**

   > *This paper presents the Grad-CAM technique. We adopted this method directly to provide class-discriminative visual explanations for our model's predictions, which is central to our goals of interpretability, error analysis, and clinical trust.*

4. **[4] F. A. Spanhol, L. S. Oliveira, C. Petitjean, and L. Heutte, "A dataset for breast cancer histopathological image classification," *IEEE Trans. Biomed. Eng.*, vol. 63, no. 7, pp. 1455–1462, 2016.**

   > *This article documents the BreaKHis dataset used in all our experiments. It provides critical information on the data acquisition protocol, magnification*

> *levels, and the label taxonomy, serving as the primary citation for our data source.*

5. **[5] A. Madabhushi and G. Lee, “Image analysis and machine learning in digital pathology: Challenges and opportunities,” *Med. Image Anal.*, vol. 33, pp. 170–175, 2016.**

   > *This survey provides essential context on the practical challenges of computational pathology, including issues of variability, validation, and explainability. It informed our decision to prioritize interpretability by including Grad-CAM and shaped our overall evaluation choices.*