Topic: Make Data Model Ready Alice Tang and Yinda Chen

TF Lite Breast Cancer Detection Week 4: Data Pre-Processing

1. Introduction

In this week's document, we are detailing the comprehensive preprocessing steps undertaken for the three datasets— dicom_data, calc_case, and mass_case— in preparation for machine learning model training. Each dataset underwent a systematic examination to ensure data quality and integrity, which involved tasks like directory adjustments, data cleaning, handling missing values, and encoding categorical variables. Additionally, we implemented tailored preprocessing techniques for various image types to effectively enhance their readiness for analysis. By implementing these preprocessing steps, our goal is to optimize our datasets for efficient modeling and analysis, positioning ourselves for favorable results in breast cancer detection.

2. Pre-Processing for "dicom_data" Dataset

The preprocessing of the dicom_data dataset is an essential step in making the data "model ready". With a thorough inspection of the basic structure of the dataset to better understand its composition, we were able to identify the necessary steps to take for cleaning. After using dicom_data.info(), we first checked for completeness of data— including identifying key attributes like missing values and redundant columns.

2.1 Directory Adjustments

Because the images are stored in a specific directory, we had to ensure that the dicom_info.csv file was able to correctly load the image data. Initially, the image paths in the dataset were stored in a directory called CBIS-DDSM/jpeg. We updated the image paths accordingly to the correction location, ../data/jpeg, which helped facilitate smooth access to the image files during further processing later. There were three distinct subsets that were created based on the SeriesDescription column.

1. **Full mammogram images:** This subset contains paths for the complete mammogram images.
2. **Cropped images:** This subset holds cropped versions of the mammogram images.
3. **ROI mask images:** This subset includes paths for the region of interest (ROI) masks.

Consequently, the paths for each image type were updated to reflect the correct corresponding directory structure.

2.2 Data Cleaning

There were multiple unnecessary columns that would not contribute to the analysis. These were removed to streamline the dataset. Columns related to personal and procedural metadata, such as

PatientBirthDate, StudyDate, SOPInstanceUID, and others, were dropped. Such columns contained metadata that were not relevant for immediate analysis, and could contain potentially sensitive information. Removing these unnecessary columns will aid in reducing the memory footprint, which is particularly crucial given the presence of large volumes of categorical variables in the other datasets. The cleaned dataset was further investigated to ensure these columns were successfully removed by rechecking the cleaned data structure with dicom_cleaning_data.info().

2.3 Handling Missing Values

Next, the dataset was examined for missing values using the isna() function. To address missing values in critical columns like SeriesDescription and Laterality, we applied the backfill method (bfill). This is a popular method of filling in missing data in medical imagery, as this ensures the missing data were filled based on subsequent valid entries in these columns— which effectively eliminates null values and maintains data integrity. With these preprocessing steps, the DICOM dataset was successfully transformed into a clean and structured format, ready for further analysis and modeling.

3. Pre-Processing for "calc_case" Dataset

To ensure the calc_case dataset would be ready for modeling, several preprocessing steps were applied to the calcification data. These steps involve data type adjustments, handling missing values, and performing one-hot encoding on categorical data.

3.1 Creation of a Working Copy

To start, we created a deep copy of the calc_train data frame and named it Data_cleaning_1. We took this step as a precaution so that any modifications made during preprocessing would not alter the original dataset.

3.2 Data Inspection

We checked the basic data frame information using Data_cleaning_1.info(), which allowed us to check which columns should be adjusted or cleaned.

3.3 Data Type Conversion

Several columns were mislabeled as objects but were categorical. We then converted these columns into the appropriate data type using astype('category'). The columns that were converted included:

- pathology
- calc type

- calc distribution
- abnormality type
- image view
- breast density
- left or right breast

Converting these variables into the correct category was paramount, as it allows for efficient memory usage and prepares the data for encoding steps.

3.4 Handling Missing Values

To handle these missing values, the isna().sum() function was employed to identify the columns with any missing data. There were 20 missing values in calc type, and 376 in calc distribution. The bfill (backward fill) method was applied to these columns, ensuring that any missing values were filled based on the subsequent values in the dataset.

3.5 One-Hot Encoding of Categorical Variables

One-hot encoding was applied to the categorical columns to transform them into binary variables, which allows machine learning algorithms to effectively interpret these features. The following columns were encoded: pathology, calc type, calc distribution, abnormality type, image, view, breast density, and left or right breast. Encoding these categorical variables in a readable format is a crucial step for later model training. The drop_first = True parameter was utilized to avoid multicollinearity in linear models by dropping the first category from each feature. In the end, there was a total of 66 columns, because there were multiple binary columns for each categorical feature applied. All of the OHE columns are “bool” type because each column represents the presence (True) or absence (False) of a particular category.

3.6 Final Data Structure Check

Lastly, the structure of the newly transformed dataframe (Data_cleaning_1_ohe) was checked again using info() to verify that the encoding was successful and the dataset was ready for model training. Thoroughly checking all preprocessing steps will ensure that the calcification case data is optimized for further analysis and model training, with all categorical variables properly encoded and missing values handled efficiently.

4. Pre-Processing for “mass_case” Dataset

Similar to the calcification case dataset, the mass case dataset underwent similar preprocessing steps to establish proper data quality and proper format for analysis.

4.1 Creation of a Working Dataset

The mass_train data frame was assigned to a new variable, Data_cleaning_2, which will serve as a working copy for preprocessing. Once again, this is done so that any transformations do not alter the original dataset.

4.2 Data Inspection

The structure of the dataset was first examined using Data_cleaning_2.info(). This allowed for an overview of the data types and the presence of any missing values. To ensure consistency with the calc_case dataset, the column breast_density was renamed to breast density using the rename() method.

4.3 Data Type Conversion

Again, there were several columns were identified as categorical and converted using astype('category') to optimize memory usage and prepare for encoding. The columns converted included:

- left or right breast
- image view
- mass margins
- mass shape
- abnormality type
- pathology
- breast density

4.4 Handling Missing Values

Missing values were identified using isna().sum(). To fill the missing values in the mass shape and mass margins columns, the bfill (backward fill) method was applied again, ensuring missing values were replaced with subsequent values in the dataset.

4.5 One-Hot Encoding of Categorical Variables

One-hot encoding was applied to convert categorical variables into binary form, facilitating their use in machine learning models. The following columns were encoded: left or right breast, image view, mass margins, mass shape, abnormality type, pathology, and breast density. The drop_first=True

parameter was employed again to prevent multicollinearity issues in linear models through excluding the first category for each encoded feature.

4.6 Final Data Structure Check

The structure of the new One-Hot Encoded dataframe (Data_cleaning_2_ohe) was checked again to confirm successful encoding and readiness for downstream tasks, such as training machine learning models. Mass case data is now optimized for further analysis, with all categorical features properly encoded and missing values addressed.

5. Pre-Processing the Image Data

Effectively pre-processing the image dataset involved several steps for consistency and to prepare them for further analysis, including resizing, contrast enhancement, and normalization. The images in the dataset are divided into three categories: full mammogram images, cropped images, and ROI mask images. Each type of image underwent a tailored preprocessing pipeline to ensure that they were ready for model training or evaluation.

5.1 Pathology Extraction from Images

The function get_pathology() was custom created to retrieve the pathology label for a given image based on the image path. The pathology labels were extracted from corresponding dataframes (Data_cleaning_1, Data_cleaning_2, etc.) depending on whether the image belonged to the calcification or mass dataset and whether it was a training or test image.

5.2. Image Display and Categorization

A show() function was designed to display sample images from each of the three categories (full mammogram, cropped, and ROI mask images) and their associated pathology labels. This function is crucial in helping verify the correctness of the images and labels before proceeding with more advanced preprocessing.

5.3 Image Resizing

Each image type was resized to a target size before further preprocessing to ensure uniform input dimensions for the later stages of analysis or in training future models.

- **Full mammogram images:** Resized to 256 x 384 pixels.
- **Cropped images:** Resized to 256 x 256 pixels.
- **ROI mask images:** Resized to 2400 x 3600 pixels.

The resize_image() function handled this resizing task using OpenCV's cv2.resize() method.

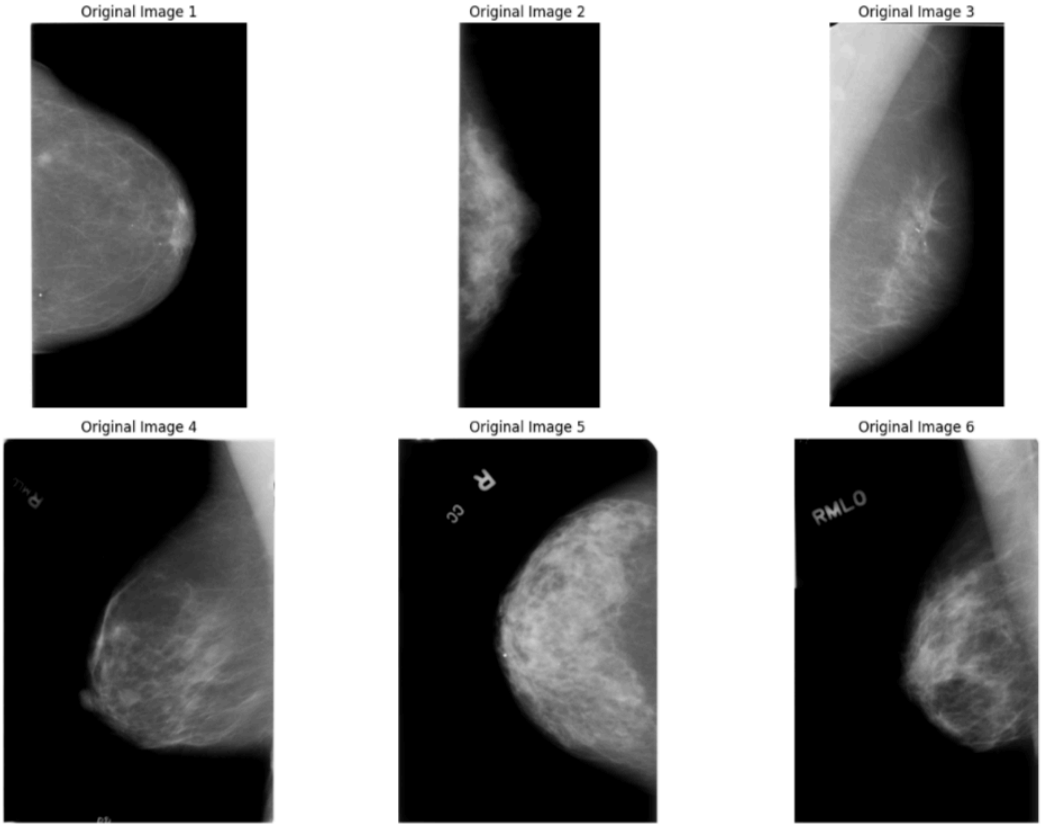
5.4 Full Mammogram Image Preprocessing

The following preprocessing steps were applied to full mammogram images:

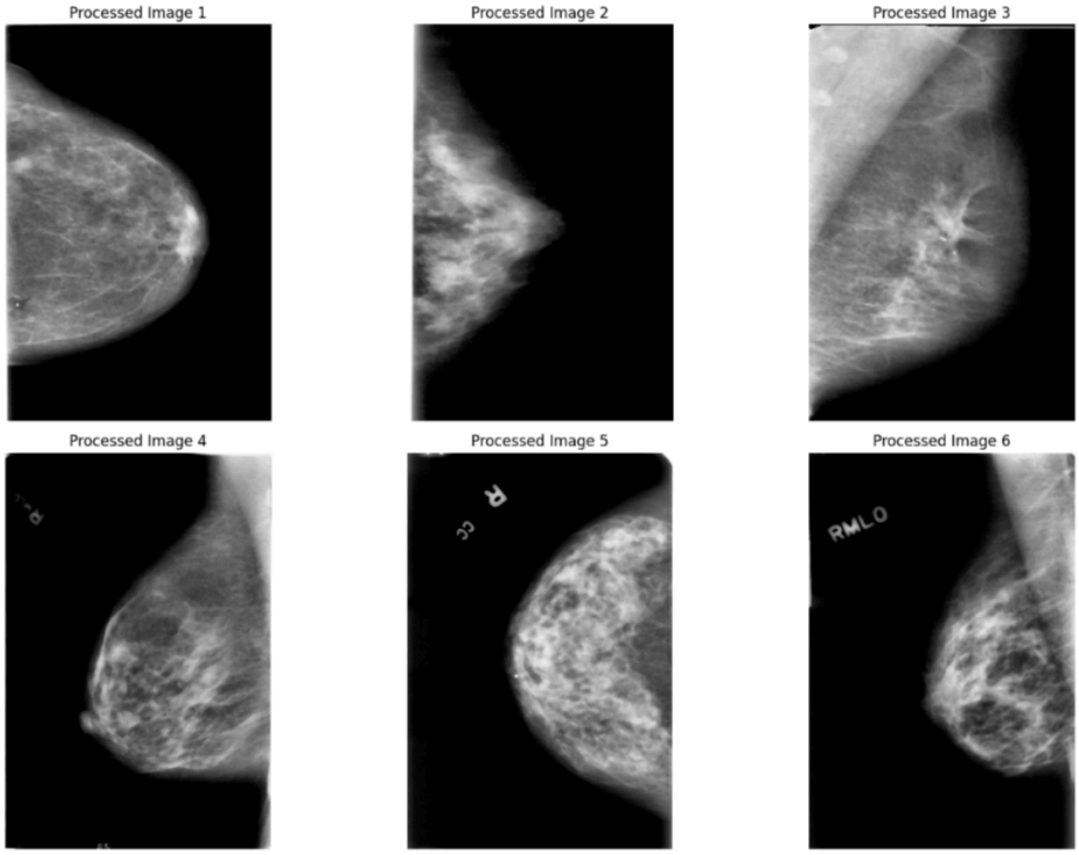
- **Grayscale Conversion:** The images were converted to grayscale using cv2.cvtColor() to remove color channels.
- **Contrast Enhancement:** CLAHE (Contrast Limited Adaptive Histogram Equalization) was applied to improve the visibility of features in the mammograms.
- **Noise Reduction:** Median blurring was applied to reduce noise while preserving edges.
- **Normalization:** The pixel values were normalized to the range [0,1][0,1][0,1] to prepare them for model input.

These steps are implemented within the preprocess_mammogram() function.

Original Full Mammogram Images:



Processed Full Mammogram Images:



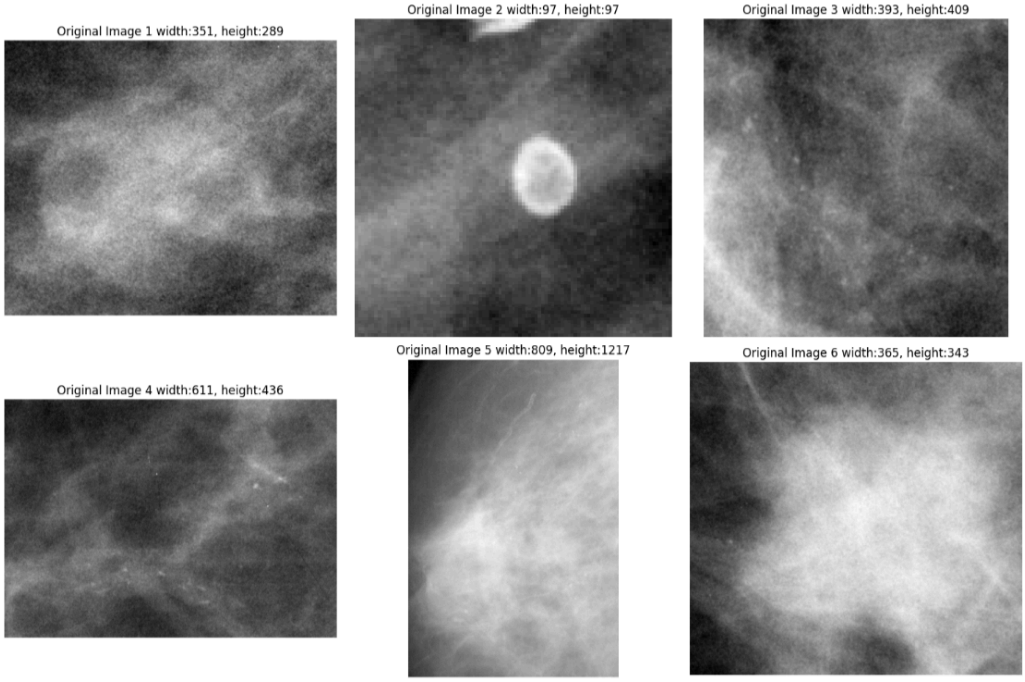
5.5 Cropped Image Preprocessing

For images that were cropped, we employed a similar, yet slightly modified pipeline:

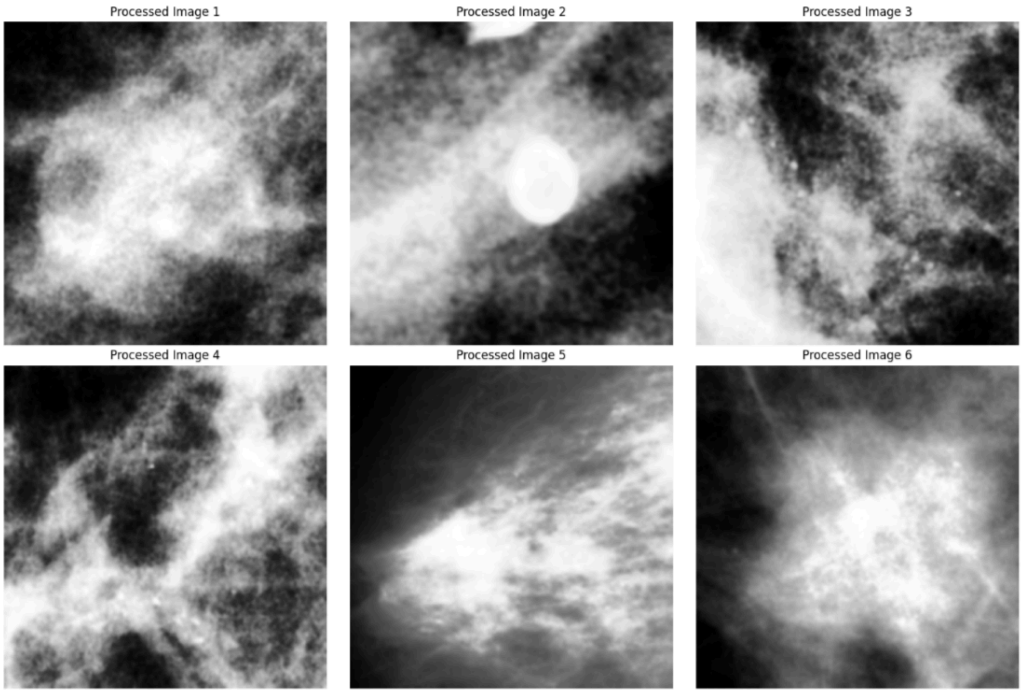
- **Grayscale Conversion:** Removed color channels to simplify the image.
- **Gaussian Blurring:** Reduced noise and smoothed out the image.
- **Histogram Equalization:** Enhanced the contrast of the image to improve the visibility of features.
- **Normalization:** Normalized the pixel values to the range [0,1][0,1][0,1].

Similarly, the function preprocess_cropped_image() performed these steps for cropped images.

Original Cropped Images:



Processed Cropped Images:



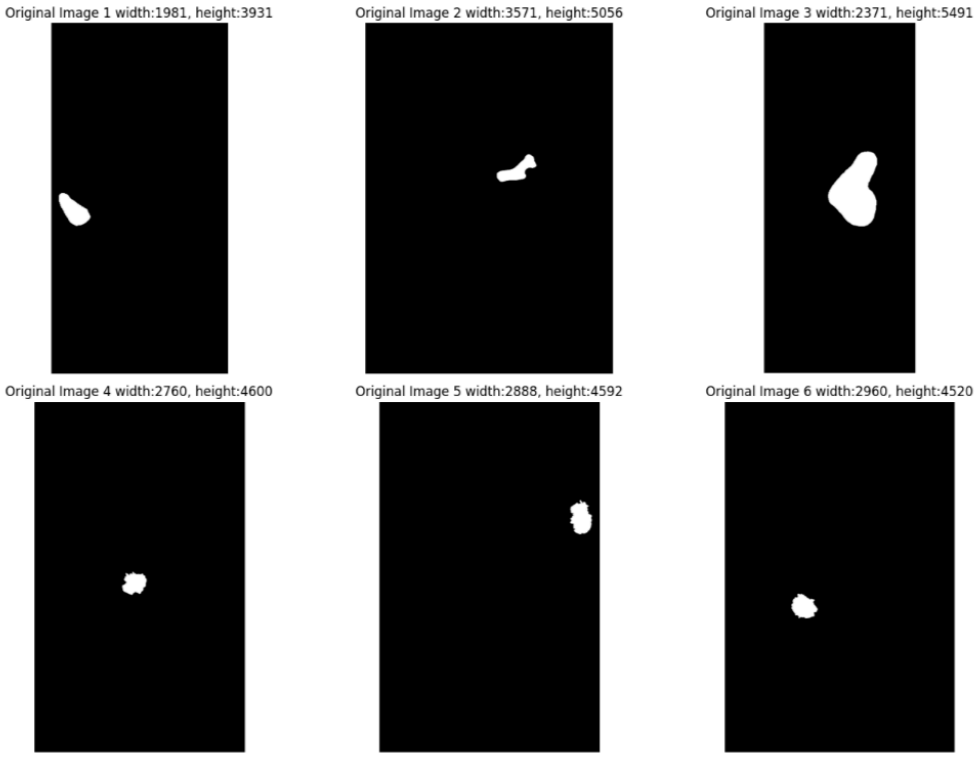
5.6 ROI Mask Image Preprocessing

The ROI (Region of Interest) mask images, which draw attention to key areas of the mammograms, underwent a customized pipeline as well:

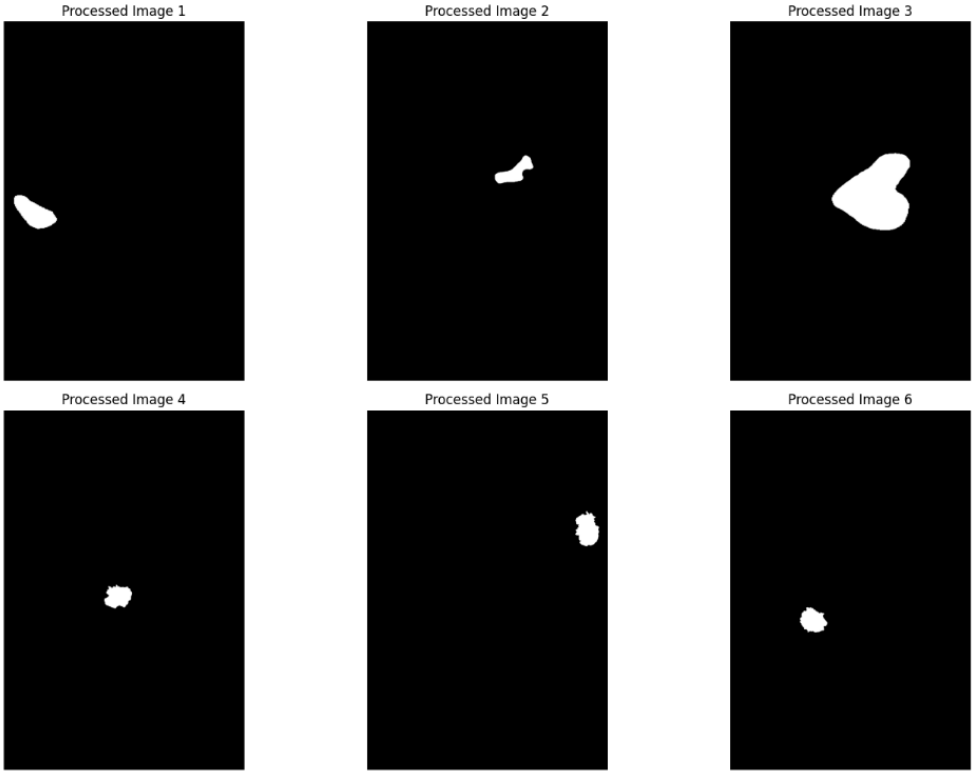
- **Grayscale Conversion:** Transformed the image to grayscale for easier binarization.
- **Binarization:** Converted the image to a binary format (pure black and white) using a thresholding method.
- **Morphological Operations:** Applied morphological opening to remove noise and clean up the binary masks.
- **Normalization:** Normalized pixel values to the range [0,1][0,1][0,1] for consistency.

The function preprocess_roi_image() managed the preprocessing of ROI mask images.

Original ROI Mask Images:



Processed ROI Mask Images:



6. Takeaways and Future Considerations

During the preprocessing phase, we learned that consistent data formatting, particularly in image paths, is crucial for correctly linking pathology data with images. Handling missing values through backfilling was effective, but future datasets could require more sophisticated imputation techniques to avoid potential bias. One-Hot Encoding (OHE) successfully transformed categorical data for machine learning, but we must continuously monitor feature dimensionality as we scale the dataset to avoid overfitting or going over memory constraints. For image preprocessing, the steps of grayscale conversion, contrast enhancement, noise reduction, and normalization proved effective for standardizing our mammogram, cropped, and ROI mask images.

Moving forward, we believe that our biggest challenges will be image size and memory constraints, especially when processing high-resolution images. We plan on using memory optimization or batch processing to maintain performance. Additionally, we may revisit our imputation strategy based on dataset characteristics, and potential class imbalances may require us to employ techniques like oversampling or synthetic data generation to ensure robust model training. We will keep these considerations in mind to ensure that our models generalize well and do not fall privy to common pitfalls encountered during the training process.