From Textures to Segmentation: Assessing Machine Learning Models for Lung Disease Prediction in Computed Tomography (CT) Imagery

Joseph Novak, Shawn Oyer, Patrick O'Brien, Thu Tran
Drexel University, DSCI 592

8/29/2024

Abstract

Lung cancer remains one of the deadliest cancers worldwide, with high mortality rates underscoring the urgent need for early and accurate diagnosis. In this study, we present a machine learning-based diagnostic toolset to discern healthy lung tissue from malignant tissue using computed tomography (CT) imagery. Our approach utilized two publicly available datasets: Iraq-Oncology Teaching Hospital/National Center for Cancer Diseases (IQ-OTH/NCCD) Lung Cancer dataset for model training/testing and the DLCTLungDetectNet dataset for validation. An image processing and UNET-based segmentation pipeline were developed, and two primary methodologies were evaluated: one utilizing raw and segmented lung scans fed into a Convolutional Neural Network (CNN), and another focusing on the extraction of texture features from segmented lung tissue for input into a Logistic Regression model. The CNN model trained on raw images exhibited near-perfect performance on the testing dataset but showed significant degradation in performance during external validation, suggesting overfitting and sensitivity to dataset differences. Conversely, the texture-based Logistic Regression model demonstrated more stable performance across both the testing and validation datasets, achieving the highest validation weighted F1 score of 80%, highlighting its potential for broader application. Our findings reveal the critical importance of image segmentation and feature extraction techniques in developing reliable diagnostic tools for prediction of lung cancer.

All code and analysis for this project are available at Tumornator's DSCI592 GitHub repository: GitHub Repo

1 Introduction

In 2024, it was estimated that lung cancer accounts for 1 in 5 US cancer deaths; In 2016, the 5-year relative survival rate for lung cancer was 28% [8]. These grave mortality rates reveal the devastating impact one of America's top killers has on its population. Practice-changing intervention is essential to bolster the health and wellness of Americans. To facilitate this, we developed a machine learning-based companion toolset for lung cancer diagnosis using CT scans.

CT scans are a cornerstone of lung cancer diagnosis, providing detailed cross-sectional images of the lungs that allow for the identification of nodules and other potential indicators of cancer. However, the manual interpretation of these images is both time-consuming and subject to variability among radiologists, leading to the potential for misdiagnosis or delayed diagnosis. This has spurred significant interest in the development of automated, machine learning-based tools that can assist radiologists by enhancing the accuracy and efficiency of lung cancer detection [9].

Institutions around the world use this technology across a variety of data types. The US National Cancer Institute clustered mutation data to detect disease subgroups and drug targets for precision medicine in diffuse large B-cell lymphoma [11]. Researchers at Albert-Ludwigs-University in Germany pioneered the use of CNN-based UNET architectures for cell segmentation [4]. At the Second Affiliated Hospital of Nanchang University in China, a tool to predict post-stroke pneumonia was developed using deep learning approaches [5]. We contribute to this growing body of literature by refining published precedent and introduce novelty by applying methods to an external validation dataset.

2 Research Objectives

Specifically, this project aims at four research objectives:

1. Build a machine learning model(s) for accurate diagnosis of lung cancer from CT scans
2. Compare the effects of image segmentation and feature extraction on model(s) performance and benchmark against automated feature extraction in deep learning model(s)
3. Determine and visualize pixels/features highly impacting model(s) classification
4. Assess the effectiveness of model(s) performance in an external validation setting

3 Data Sources

All data used were sourced from Kaggle and freely available under CC0 1.0 license. The first data source called the Iraq-Oncology Teaching Hospital/National Center for Cancer Diseases (IQ-OTH/NCCD) was used as a training and testing dataset that can be downloaded here: IQ-OTH/NCCD download. The dataset is based on 110 cases collected in 2019 varying in gender, age, educational attainment, area of residence, and living status [1]. It contains 1102 images in JPG format divided into separate directories: Malignant, Benign, and Normal.

The second data source called the DLCT Lung Detect Net was used as a validation dataset that can be downloaded here: DLCTLungDetectNet download. The dataset is a comprehensive and curated collection of diverse datasets related to lung tumors and pulmonary conditions [2]. It contains 1000 images in JPG and PNG format divided into separate directories: Adenocarcinoma, Large Cell Carcinoma, Squamous Cell Carcinoma, and Normal.

4 Pre-Processing

4.1 Image Pre-processing / Segmentation

Given the large number of scans from a variety of different subjects and hospitals, pre-processing each image was an important step in creating consistency within our datasets. Beginning with image coloring, we applied a grayscale filter to each scan in order to view each scan through the same color lens. This minimized any potential coloring variations between images that could be a result of different lighting or machinery used at the time each scan was taken. Orientation of each image was also standardized so that each image would consistently show the same position of subject in the scan.

The next steps of pre-processing focused mainly on normalizing the quality of each image as well as reducing noise within all scans. This was done through the use of bitplane slicing and Gaussian blur. Bitplane slicing essentially takes an 8-bit image and creates 8 planes, each representing the pixel intensity from a given bit. The lower bitplanes are considered the least significant planes and typically contain the most noise. The higher bitplanes are most significant and represent the general image structure, shape, and largest intensity variations of the image.

The lower bitplanes were discarded and the image was reconstructed using the remaining planes. These steps resulted in an image with reduced noise and more emphasis on the significant characteristics of the image. Gaussian blur was the final step in image processing and was used to suppress noise and improve edge detection for later image segmentation.

With a new level of consistency between images applied across the scans after our pre-processing steps, segmentation was applied in order to isolate the specific area of interest within each scan. Specifically, we segmented the image to highlight the areas within the lungs and mask the remaining portions of the scan. This is an important step because it allows our modeling to focus exclusively on the areas within the lungs, where markers indicating any malignancies are likely to be found. This process also removes potential unwanted biases that could be found in the backgrounds of these images, unrelated to the lungs.

Initially, the image segmentation was performed using the skimage morphology library. This involved creating a processing pipeline that performs a variety of operations to the scan, before producing the segmented output. These operations included binarization, erosion, dilation, and removing or filling holes in the binary image to eventually obtain a mask that covers everything besides the inner lung areas. Although this process was fairly successful in segmenting most images in our main data source, it was not a surprise since the pipeline was specifically created and tailored to the scans in this dataset. This lack of pipeline versatility or over specificity became clear when this pipeline was applied to scans in the validation dataset and struggled to correctly segment images at the same level as it had in the main dataset.

Although the skimage pipeline was only partially successful in creating a generalized approach to segmentation, it fortunately provided us with a valuable training dataset for a UNET segmentation model.

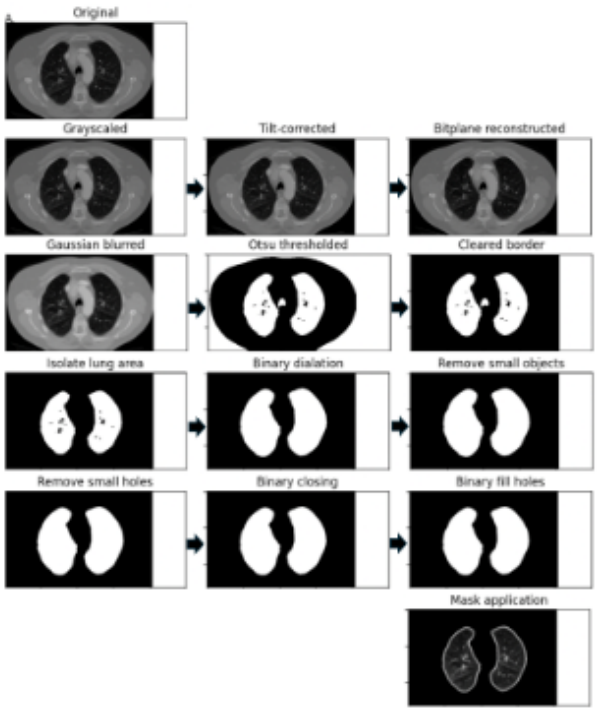

Figure 1: Skimage Segmentation Pipeline

Using the outputs from the initial pipeline, we were able to select cases of high quality segmentations and save out the original image as well as the applied mask. Around 400 image-mask pairs were selected for this modeling. These image-mask pairs were used in the training of a UNET segmentation model, with the mask being the ground truth label for the given input image. The resulting trained model was able to take an image input and produce an output mask resulting in a segmented image when applied to the original. Once again, this approach was largely successful in segmenting the main dataset. When applied to images in the validation dataset, the quality of the segmentation declined, but to a lesser extent than the validation segmentation using the original pipeline. A potential explanation of this decline in UNET segmentation performance could lie in the general differences between dataset types. Specifically, the dimensions in the validation dataset were not uniform, whereas the main dataset image dimensions were almost entirely 512x512. This causes problems in UNET architecture since convolution and pooling operations require fixed input sizes. This meant the majority of validation images needed to be resized to fit the 512x512 standard and could have negatively affected the performance of the UNET segmentation.

Through the use of image pre-processing techniques as well as image segmentation, we were able to create a more consistent and focused lung CT scan dataset. These steps play a crucial role in enhancing performance of both the logistic regression and CNN models. By standardizing the images and focusing on the lung regions, we reduced noise and variability that could negatively affect model accuracy.

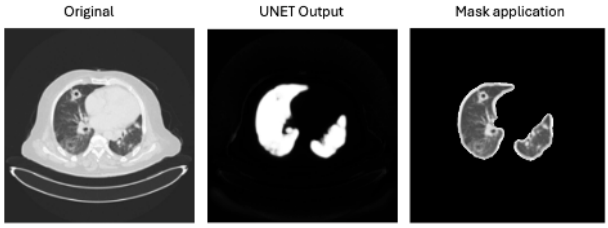

Figure 2: UNET Segmentation

4.2 Feature Extraction

Extracting features in an image quantifies the frequency of different combinations of neighboring pixel values. Neighboring pixels in a 2D image can be analyzed in four cardinal directions: 0, 45, 90, and 135. The combination of directional analysis gives an average Gray Level Co-Occurrence Matrix (GLCM) that depends less on factors such as the bed dip and azimuth. Eichkitz's article also gave a more detailed workflow on how the method GLCM works in texture analysis, figure 3. First, the picture is converted into a grayscale image. This grayscale image is then transformed into a matrix where each element corresponds to the intensity value of each pixel. This matrix will then serve as a base for further GLCM calculations. Those calculations are describing relations between values of pixels at some offset given by the parameters distance and angle. The GLCM obtained in such a manner provides an indication of the frequency with which pairs of pixel intensities occur in a certain spatial relationship in the image. Then, various orientations like 0, 45, 90, and 135 are computed to get features for texture evaluation, such as Energy, Contrast, Homogeneity, Entropy.. The computation of the GLCM in all directions gives a general feature that represents the texture in a more complete way, including more spatial relationships [3].

Figure 3 part i provides further explanations of the definitions and formulas for some of the key features of GLCM textures: Angular Second Moment-ASM, or Energy, which characterizes the uniformity of the texture. The higher the value, the more homogenous the texture is. Contrast measures the variation of intensity on a local scale. Correlation refers to a lin-

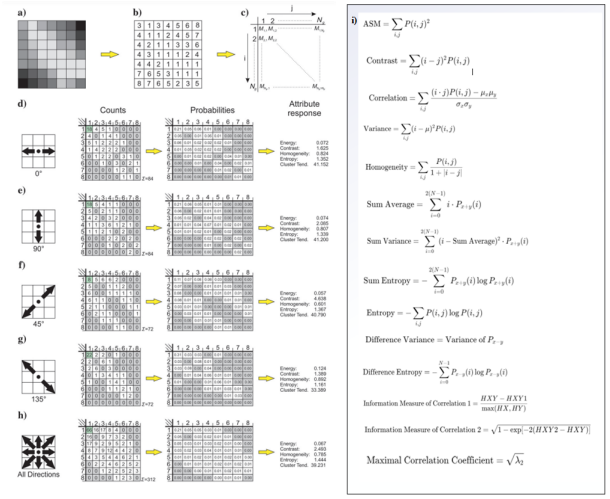

Figure 3: GLCM Feature Approach and Formula

ear dependency between values of neighboring pixels. Other features calculated are Variance, Homogeneity, Sum Average, Sum Entropy, and Difference Entropy. These features give additional information about the complexity and structure of the texture. Such measurements are useful in different applications, including texture classification, image segmentation, and analysis, which may be applied to medical imaging or remote sensing. A workflow is hereby presented that summarily details how the GLCM method can be applied in exploiting the spatial relationships among the intensities of pixels towards the derivation of meaningful features of textures which could be useful in image processing tasks.

After researching the GLCM theory, we loaded directories containing images and transferred metadata into a pandas dataframe called 'Lung_df'. Then, the images were processed through OpenCV's cv.imread function. It prepares the images to normalize them onto a comparable scale for the analysis of texture features, applying Z-score normalization for standardizing the pixel values across the dataset. This step enhances robustness in feature extraction and ensuing modeling. Then, the code performs feature extraction using the GLCM method on the processed images for texture features. Finally, given that the algorithm will extract a large number of patches, the analysis excludes zero-valued background pixels by using the function pyfeats.glcm_features with the additional argument ignore_zeros=True.

Figure 4 presents two tables of the feature statistics of GLCM for two segmentation methods, namely: Table 1 for Scikit Segmentation and Table 2 for UNET Segmentation. Each table is devoted to several statistical measures of different features of GLCM such as Angular Second Moment (ASM), Contrast, Correlation, etc. Both lists show count, mean, std, minimum value, percentiles at 25%, 50% (median) and 75%, maximum value for each feature. In Table 1, concerning Scikit segmentation, the average of GLCM_ASM is 0.298401, with a standard deviation of 0.07638 and a maximum value of 0.937620. It therefore denotes the fact that Scikit segmentation has captured texture patterns in a manner which is moderately uniform. However, GLCM_Contrast recorded a mean of 0.215507, hence showing less variance in variation within pixel intensities. GLCM_Entropy, in Scikit segmentation, recorded a mean of 2.090601, which evidences greater randomness in the texture pattern. By contrast, the UNET segmentation table in Table 2 reveals very different results. The mean value of GLCM_ASM is considerably higher, at 0.556702, which shows that UNET captures more homogeneous texture patterns. GLCM_Contrast is much higher, with an average of 2142.13, indicating that UNET segmentation does capture more intensity variation across pixels. Also, the GLCM_SumOfSquaresVariance shows large variability with a mean value as high as 5967.059716. However, its value is less with 1.326075 for the GLCM_Entropy, indicating more structured and less random textures when compared to Scikit segmentation.

Table 1: Scikit Segmentation GLCM Feature Statistic

	count	mean	std	min	25%	50%	75%	max
GLCM_ASM	1100.0	0.298401	0.076388	0.171366	0.261847	0.287581	0.318378	0.937620
GLCM_Contrast	1100.0	0.215507	0.046995	0.000000	0.198942	0.223798	0.247000	0.313437
GLCM_Correlation	1100.0	0.903077	0.152256	0.527413	0.735691	0.999715	0.999961	1.000000
GLCM_SumOfSquaresVariance	1100.0	2175.375223	3513.600325	0.053878	3.122760	389.209753	2816.053115	15592.076551
GLCM_InverseDifferenceMoment	1100.0	0.892422	0.023369	0.844598	0.876778	0.888217	0.900550	1.000000
GLCM_SumAverage	1100.0	23.920881	36.953521	2.072013	3.437720	6.349602	26.333954	212.929421
GLCM_SumVariance	1100.0	8701.285384	14054.419567	0.184988	12.268060	1556.685092	11264.023364	62368.162420
GLCM_SumEntropy	1100.0	1.874217	0.273049	0.234650	1.748603	1.900980	2.032368	2.608907
GLCM_Entropy	1100.0	2.090601	0.305576	0.249392	1.963959	2.132219	2.264556	2.884432
GLCM_DifferenceVariance	1100.0	0.023137	0.040528	0.002255	0.002493	0.002623	0.003102	0.179717
GLCM_DifferenceEntropy	1100.0	0.732741	0.110747	0.000000	0.712184	0.759248	0.797807	0.887937
GLCM_Information1	1100.0	-0.402681	0.120514	-1.000000	-0.470617	-0.365814	-0.313037	-0.205331
GLCM_Information2	1100.0	0.783542	0.088055	0.387265	0.719529	0.771211	0.853323	0.957846
GLCM_MaximalCorrelationCoefficient	1100.0	1.107289	0.191258	0.000000	1.031139	1.171604	1.206546	1.344794

Table 2: Unet Segmentation GLCM Feature Statistic

	count	mean	std	min	25%	50%	75%	max
GLCM_ASM	1102.0	0.556702	0.144283	0.205620	0.467658	0.550375	0.646624	0.968214
GLCM_Contrast	1102.0	2142.136141	500.809280	0.000000	2021.465182	2229.738607	2403.802067	4046.051947
GLCM_Correlation	1102.0	0.807295	0.056143	0.714911	0.767927	0.793772	0.826073	1.000000
GLCM_SumOfSquaresVariance	1102.0	5967.050916	2113.864892	1004.104343	4730.480289	5493.654123	6430.752177	16122.422547
GLCM_InverseDifferenceMoment	1102.0	0.942783	0.019507	0.858159	0.933576	0.942865	0.952425	1.000000
GLCM_SumAverage	1102.0	57.277942	30.229038	10.037706	43.104080	50.304786	59.719788	393.343217
GLCM_SumVariance	1102.0	21726.067525	8447.160587	4016.417123	16671.607828	19620.580305	23343.172586	64488.189776
GLCM_SumEntropy	1102.0	1.326075	0.388433	0.121518	1.114789	1.367390	1.563024	2.499954
GLCM_Entropy	1102.0	1.407702	0.419271	0.121769	1.179675	1.446890	1.655746	2.707242
GLCM_DifferenceVariance	1102.0	0.003305	0.000229	0.002434	0.003198	0.003310	0.003416	0.003891
GLCM_DifferenceEntropy	1102.0	0.488682	0.161202	0.000000	0.424827	0.501874	0.576145	1.034153
GLCM_Information1	1102.0	-0.578577	0.092112	-1.000000	-0.601706	-0.559673	-0.527870	-0.362606
GLCM_Information2	1102.0	0.805840	0.082202	0.456892	0.768992	0.831148	0.860603	0.946170
GLCM_MaximalCorrelationCoefficient	1102.0	1.066830	0.209789	0.000000	1.065354	1.128896	1.157266	1.329319

Figure 4: GLCM Feature Statistics

In summary, UNET segmentation generally offers higher values for GLCM features of ASM, Contrast, and Sum of Squares Variance; thus, it is capturing de-

tailed variations in texture. On the other hand, Scikit segmentation tends to identify complicated and random patterns in texture, reflected by the high entropy measured. Differences suggest the relative strengths of both segmentation approaches with respect to textural analysis.

5 Exploratory Data Analysis and Data Visualization

The Exploratory Data Analysis (EDA) presented in Figure 5 offers a detailed comparative analysis between Scikit Segmentation and UNet Segmentation across both test and validation datasets. In this systematic analysis, each of the segmentation methods is compared section-wise, which mainly includes the following: GLCM feature histograms, box plots, correlation heatmaps, and PCA plots. This structure allows for a closer comparison of how each method performs in feature extraction and analysis.

The GLCM Feature Histograms section starts off the analysis with Scikit Segmentation in the test set (Fig 5.1). Different histograms are given according to different GLCM features such as ASM, which stands for Angular Second Moment, Contrast, Correlation, SumOfSquaresVariance, amongst others. In the GLCM_ASM histogram, values lie between 0.2 and 0.6, indicating that the textures have a regular character with moderate uniformity. The GLCM_Contrast feature shows that all the textures are smooth, and their contrast values are low, mostly lying in the range of 0.1 to 0.3. The higher values of the concentration are observed to lie around 0.7 to 0.8 in the case of GLCM_Correlation, and it is representative of repeating patterns in segmented regions. The histogram of the GLCM_SumOfSquaresVariance is highly skewed toward low values, with a few samples reaching very high values; thus, it is expected that the variability in the test set captures a lot of variation in the textural complexity. Values of GLCM_Entropy vary from 1.5 to 2.5-the randomness in textures is moderate in this dataset.

The histogram of GLCM features for the Scikit Segmentation validation set represented in Fig 5.2 has a similar distribution as that of the test set, hence consistent feature extraction. The distributions of GLCM_ASM, GLCM_Contrast, and GLCM_Correlation are consistent, but GLCM_SumOfSquaresVariance and GLCM_Entropy continue showing skewed distributions, an indication that the validation set also possesses similar texture complexities and randomness as test set. For UNet Segmentation, GLCM feature histograms are more concentrated around 0.5 in the test set (Fig 5.3), which indicates that there is more consistency in texture uniformity. Also, GLCM_Contrast has a much narrower distribution, showing less variability. GLCM_Correlation is even more peaked near 0.85, reflecting that UNet captures more consistent repeating patterns in the segmented regions. In addition, GLCM_SumOfSquaresVariance still remains very skewed but with a greater concentration of values towards the low variance level, meaning lesser complexity in texture. The range in GLCM_Entropy is closer, between 1.5 and 2.0, representing a more consistent entropy level for the test set. In general, GLCM Feature Histograms for UNet Segmentation of the validation set are very similar to those of the test set shown in Fig. 5.4. This basically means that UNet Segmentation consistently yields the same features across different sets. From the consistent distribution in GLCM_ASM, GLCM_Contrast, GLCM_Correlation with the skewed but stable GLCM_SumOfSquaresVariance and GLCM_Entropy, it can be affirmed that UNet Segmentation is a reliable and consistent way of feature extraction.

In the GLCM Feature Box-Plots section, Scikit Segmentation's boxplots for the test set (Fig 5.5) depict that in malignant samples, the median ASM values are slightly lower compared to non-malignant samples; however, there was a significant overlap in the interquartile ranges which means that this feature alone might not help identify the two classes effectively. Similar distributions of both classes in GLCM_Contrast represent low values of contrast and some outliers, hence their poor discriminatory power. GLCM_Correlation shows very high medians for both classes with high overlapping and some outliers in a non-malignant class. For the GLCM_SumOfSquaresVariance, one could notice a wide range of values, where usually the non-malignant samples have higher variability, hence varied texture complexity. In addition, GLCM_Entropy gives lower values for malignant samples; however, the possible overlap between classes may indicate other features that are necessary to achieve better classification accuracy.

The Box-Plots for the GLCM Features for the Scikit Segmentation Validation set, Fig 5.6, present consistent characteristics with that of the test set, thus serving as an indicator of reliable feature extraction across datasets. The feature GLCM_SumOfSquaresVariance still presents a large range of values with many outliers, especially for the non-malignant class. For GLCM_Entropy, somewhat similar trends are observed with slightly lower entropy values for malignant samples. Box-plots

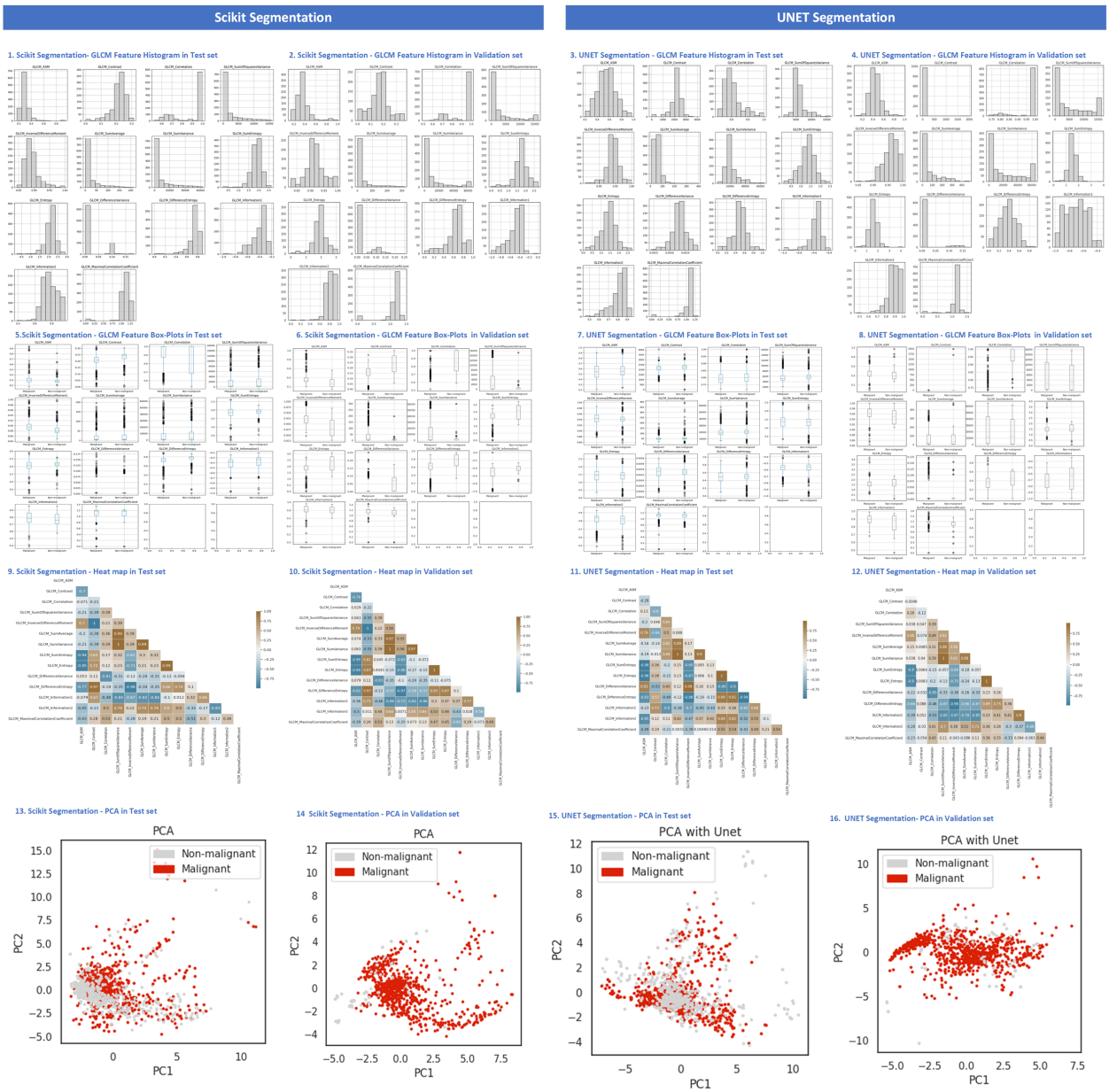

Figure 5: EDA for Scikit & UNET Segmentation in both Test & Validation Sets

for feature GLCM of UNet Segmentation test set (Fig 5.7) are more clustered for malignant and non-malignant samples, which means fewer outliers for the feature extraction. Similarly, GLCM_Contrast is more concentrated. Overlap between two classes is still there but less variable than Scikit Segmentation. GLCM_Correlation has a higher median in both classes, with fewer extreme outliers, which shows that UNet captures correlation in a more consistent manner. The distribution of the GLCM_SumOfSquaresVariance and GLCM_Entropy is more compact for UNet Segmentations with fewer outliers; this means that this method provides more consistent measurements. The Box-Plots for the GLCM Features of the validation set UNet Segmentation (Fig 5.8) confirmed the findings derived from the test set and thus stability in the features extracted. The more concentrated distributions of the GLCM_ASM, GLCM_Contrast, and the GLCM_Correlation and overall, the more compact distributions of GLCM_SumOfSquaresVariance and GLCM_Entropy, which state the UNet Segmentation

copes with the variability.

In the heat maps that display the correlations, there exists a strong positive correlation in the Scikit Segmentation test set heatmap, Fig 5.9, between the GLCM_SumVariance versus the GLCM_SumAverage with a value of 0.99, which may indicate feature redundancy. Strong negative features are seen in GLCM_Entropy versus GLCM_ASM, at -0.94, which may indicate that as there is more uniformity in texture there is less randomness. Moderate features exist in the data as can be seen between GLCM_MaximalCorrelationCoefficient and GLCM_Information2 with a value of 0.66 indicating features that relate to each other but are non-redundant. Below, the heatmap from the validation set for the Scikit Segmentation, Fig 5.10 is representative, showing consistent patterns of correlation compared to the test set. Most of these are consistently correlated, though one or two are weakly different in magnitude-for example, the correlation of GLCM_Entropy and GLCM_SumEntropy is very slightly weaker compared to the test set. The UNet Segmentation test set heatmap (Fig 5.11) overall shows stronger correlations across many feature pairs compared to Scikit Segmentation. For instance, the negative correlation between GLCM_Entropy and GLCM_ASM is greater (-0.98), which would imply a stronger negative relationship between.The UNet Segmentation heatmap of the validation set, Fig 5.12, reveals a pattern similar to that of strong positive and negative correlations in the test set, reflecting stability of feature relationships across various datasets.

As evident from the PCA plots in this section, both segmentations poorly classify the non-malignant from the malignant samples using the first two PCs. For example, looking at the Scikit Segmentation test set PCA plot shown in Fig 5.13, the malignant and non-malignant samples are hugely overlapping - an indication that the features extracted from those samples don't provide very clear class separation. Notice also how the data points are wide apart along both PC1 and PC2, pointing to variability in feature representations. Performing PCA on the validation set in Scikit Segmentation, Fig 5.14 shows similar data spread and overlap, but there are malignant samples more spread along PC1, with slight variability in feature representation. In the test set PCA plot of UNet Segmentation, Fig. 5.15, there is considerable overlap, though separation seems slightly improved compared to Scikit Segmentation. That is to say, most of the malignant samples have been stretched further along PC1, therefore UNet Segmentation has captured features that are more distinctive. We can also see the same improvement for the validation set of UNet Segmentation in Fig. 5.16, however the malignant samples are still stretched out along the horizontal axis, reflecting greater variability within the validation set.

Most of the features of GLCM are tighter and more consistent, generally indicating the better capture of uniform texture across datasets. Box plots present fewer extreme outliers for UNet Segmentation, hence proposing better handling of variability in the data. Stronger correlations in UNet Segmentation, from the heatmaps, indicate a more robust capture of feature relationships. PCA plots show that none of the methods could manage to separate classes, but slightly better separability may be seen in the UNet segmentation case, which means more discriminative features are provided by it. However, further optimization could be needed to achieve a reasonably good result.

6 Methodology

We designed a Workflow Diagram (figure 6) on the data processing pipeline with regard to segmentation, feature extraction, and model building. In looking at the reduced diagram, starting with raw data, the data undergoes cleaning, normalization, and enhancement in treatment processes, thus ensuring that quality standards are met as necessary for further analysis. The pre-processing steps are very critical since they ensure that the format is standardized, thus making the other parts of the workflow more effective and efficient.

After pre-processing, the workflow splits into two parallel segmentation pipelines. The first is UNET Segmentation. It was designed for image segmentation, especially for medical images by using neural networks in a recognized architecture. The second pipeline is done exploiting Scikit Segmentation, which adopts segmentation techniques from the Scikit-learn library- a widely used Python machine learning toolkit. In the two pipelines, both segmentation methods are further normalized using Z-scores according to the mean and standard deviation such that features are on a comparable scale in both pipelines. The extracted features through GLCM of the segmented images are further put under normalization procedures. The GLCM technique is one of the popular methods for texture analysis that can compute the spatial relationship of different gray intensities of pixel and hence most advantageous in detecting diseases like cancer in medical imaging.

The workflow then goes on to split into two paths in the case of traditional and advanced machine learning

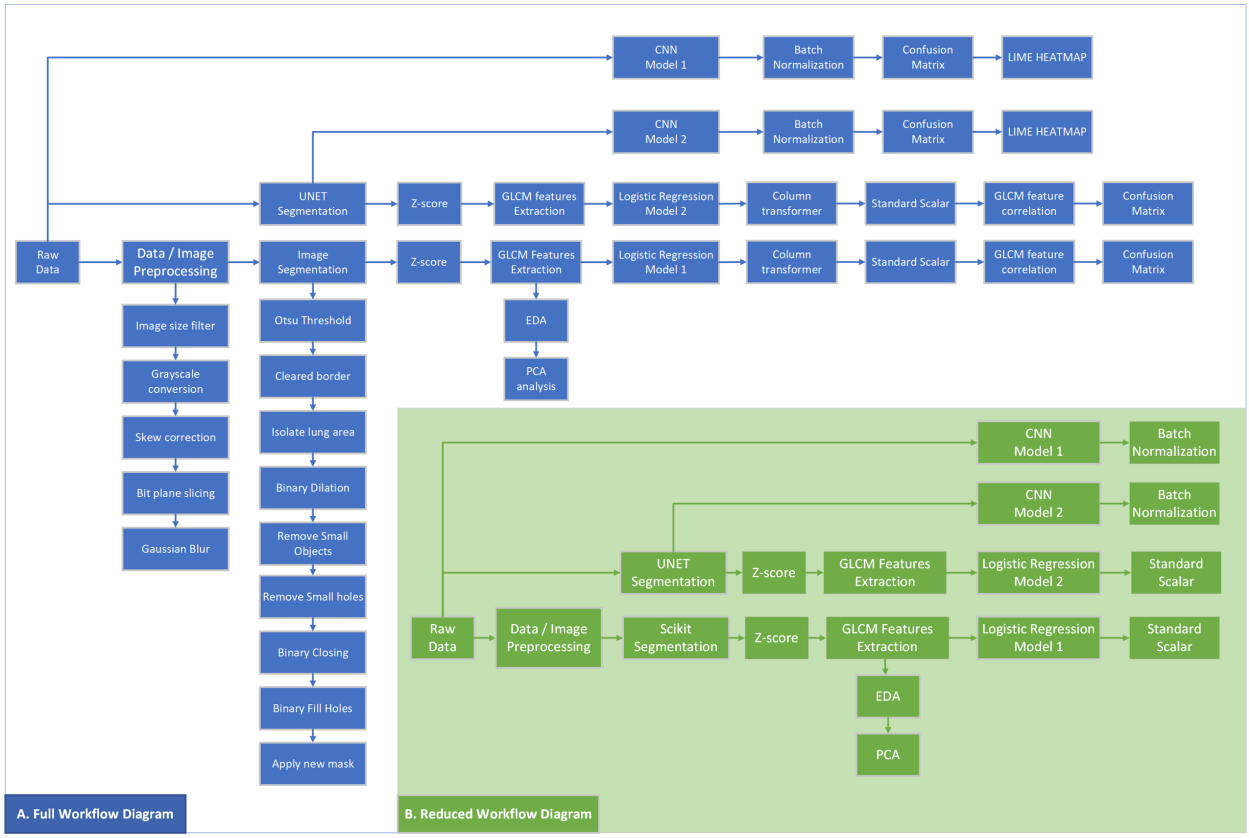

Figure 6: Workflow Diagram

models. Some of those paths lead to logistic regression models: Model 1 and Model 2 are followed by standard scaling in order to get better models. Logistic regression is one of the popular statistical models for tasks based on binary classification. Other paths lead to individual paths with CNN models: Models 1 and 2 lead to batch normalization after which stabilizing and accelerating deep learning of the models happens. Additionally, EDA and PCA at different steps of the workflow will help in understanding the data, summarizing the results, and making the data less dimensional, which, therefore, would assist in modeling.

The full workflow diagram depicts a systematic curve of a combination between conventional machine learning and advanced deep learning techniques, which assure a dynamite, robust, and interpretable analysis pipeline. To help explain and validate model predictions, key evaluation tools, such as confusion matrices and LIME heatmaps, are also integrated into the results, not only for accuracy but for transparency.

6.1 Logistic Regression Model

In the first approach, pre-processing and segmentation was applied to raw images using skimage morphology toolset. In the second approach, preprocessing was applied and segmentation performed using the newly trained UNET model. These are the procedural steps used for the Logistic Regression Model:

1. Masked areas of each scan were replaced with NA
2. Image arrays were individually z-scored
3. GLCM texture features were extracted from the remaining values
4. Split data into training and testing sets (70%/30%)
5. Features were re-distributed using the sklearn Standard Scaling method
6. A grid Search was performed on the training data to optimize model penalty and resulted in a C parameter of 10

7. A logistic regression model was trained with l1 penalty

8. The model was applied to the test set and the validation set

9. Accuracy, precision, recall, and weighted F1 score were generated for performance evaluation of each model.

6.2 CNN Model

In pre-processing the images for input into the CNN model, we created individual directories for the IQ-OTH/NCCD images to be used for training/testing and the DLCTLungDetectNet images to be used for validation. We acquired code from kaggle that helped create the custom CNN model that was used for all phases [6]. These are the procedural steps used for the CNN Model:

1. Create a new data frame containing filename (to be used as input features), labels (to be used as actual labels), data format and image metadata

2. Remove rows with bad dimensions to ensure consistency

3. Load images from file paths, resize images, normalize pixel values, flatten images, and return image arrays for each file path

4. Split data into training and testing sets (70%/30%)

5. Implement the file paths and labels into the Image Data Generator that creates the CNN pipeline

6. Set up CNN Model Sequential with all necessary parameters, compile the model and output the summary

7. Train the model on training images, save the model and history, evaluate the model, predict probabilities for validation data and evaluate metrics (Accuracy, Precision, Recall, and weighted F1)

In answering our 3rd research objective, to interpret the model and determine which pixels/features that hold the most value in making accurate predictions of malignant vs non-malignant lung tissue, we used a technique called LIME (Local Interpretable Model-Agnostic Explanations). LIME is a technique that provides the "why" behind predictions and is a look under the hood of black-box models with a local, interpretable model to explain each individual prediction. It helps identify potential issues such as information leakage, bias, robustness, and causality. It quantifies and visualizes the pixel contributions across images to determine which pixels have the highest explanation for predicting the classifier [7]. We ran LIME on both the raw images and segmented images and it works by selecting the class with the highest explanation weight, then maps each weight to the corresponding pixel, then accumulates a heatmap that visualizes the average contribution per pixel. Blue means it contributes positively towards the prediction, white means it does not contribute, and red means it contributes negatively (Figure 7).

In reviewing the accumulated heat maps, the raw images have higher contributing pixels with a slight skew to the left side of the lung, suggesting the model is picking up a pattern on the left side of the lung that is helping it predict whether its malignant or not. For the segmented image, there is a more balanced pixel weight within the lung, however, there seems to be negative contributions in the center part of the lung suggesting the model detected features or patterns that were contradictory to the final prediction in this region. We also created visualizations for how LIME was used on individual images, and what parts of the images had higher contributing pixels towards its prediction. In these images, darker red means it contributed positively towards the prediction, lighter red means a small contribution and no color means no contribution. In reviewing the segmented images, the model seemed to use black areas outside the lung as part of its contribution. We are unsure as to why that is but we found LIME to be a great tool to help us understand what were the most important pixels the model used towards accurately classifying malignant vs non-malignant images.

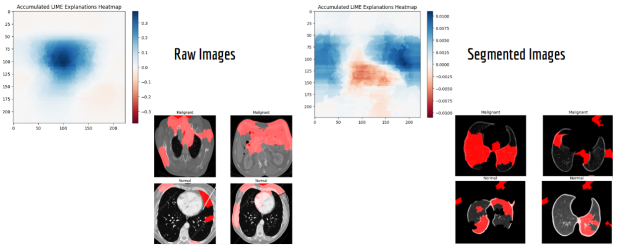

Figure 7: LIME Accumulated Heatmap and Individual Examples

Table 1: Performance Metrics Comparison of Models Across Testing and Validation Phases

	Model	Logistic_Regression_1	Logistic_Regression_2	CNN_1	CNN_2
	Segmentation Method	Scikit	UNET	None	UNET
	Input Datatype	GLCM Features	GLCM Features	Image array	Image array
	Train Source	IQ-OTH/NCCD	IQ-OTH/NCCD	IQ-OTH/NCCD	IQ-OTH/NCCD
	Test Source	IQ-OTH/NCCD	IQ-OTH/NCCD	IQ-OTH/NCCD	IQ-OTH/NCCD
	Validation Source	DLCTLungDetectNet	DLCTLungDetectNet	DLCTLungDetectNet	DLCTLungDetectNet
	#Train	770	771	770	771
	#Test	330	331	330	331
	#Validation	969	977	1000	1000
Test Malignant	Precision	81%	62%	99.70%	98%
	Recall	82%	66%	99.70%	99%
Test non-Malignant	Precision	81%	62%	100%	99%
	Recall	80%	59%	99%	98%
Test Summary	Loss	NA	NA	0.009	0.04
	Accuracy	80.90%	62.20%	99.70%	98.50%
	F1	81%	62%	100%	98%
Val Malignant	Precision	85%	77%	77%	79%
	Recall	94%	61%	91%	98%
Val non-Malignant	Precision	62%	15%	3%	28%
	Recall	36%	28%	1%	2%
Val Summary	Loss	NA	NA	2.14	0.9
	Accuracy	81.70%	54.40%	71.40%	77.70%
	F1	80%	59%	66%	70%

7 Results

This section presents the performance evaluation of a logistic regression model and a CNN model, developed for classifying lung tissue as malignant or non-malignant. All models were trained and tested on the same datasets, with varying input data types and pre-processing strategies. The models were then validated on an independent validation dataset to assess their generalizability. View Table 1 for the performance metrics comparison of models across testing and validation phases.

7.1 Logistic Regression Model Evaluation

The model, Logistic_Regression_1, trained on Scikit images, had a weighted F1 score of 81% in the test set and 80% in the validation set. Logistic_Regression_2, trained on UNET images, exalts a weighted F1 of 62% in the test set and 59% in the validation set. It is clear that the performance of Logistic_Regression_1 triumphs over that of its UNET segmented counterpart, Logistic_Regression_2.

The UNET segmentation in many instances includes areas outside of the lungs in the regions of interest. Including those non-lung regions has a detrimental effect on regression model performance. Observed GLCM features between the two segmentation methods appear different, suggesting that the origin of this effect stems from including textural features outside the lung in texture feature extraction.

Logistic_Regression_1 exemplified a satisfactory performance on the validation set suggesting this texture-based model could be generalized across datasets from different batches. Cleaner segmentation of the validation images can potentially improve the performance of this model as many of the validation images were incorrectly segmented using the skimage morphology approach.

Taken together, these results support the use of textural features in the diagnosis of lung cancer from CT scans.

7.2 CNN Model Evaluation

CNN_1, trained on raw image arrays, demonstrated near perfect performance on the test set with an accuracy of 99.70%, a precision of 99.70%, and an F1-score of 100%. The minimal loss value of 0.009 further indicates that the model fits the training data exceptionally well.

However, when validated on the DLCTLungDetectNet dataset, CNN_1's performance dropped significantly. The accuracy decreased to 71.40%, and the loss value rose to 2.14. Precision and recall values also dropped to 61% and 71%, respectively, leading to an F1-score of 66%. A closer examination of the validation results reveals a significant disparity in the model's performance between the two classes.

The model performed well in identifying malignant cases, correctly classifying 91% of them, but it struggled significantly with non-malignant cases, achieving only 3% precision and 1% recall. The substan-

tial difference in performance between malignant and non-malignant classifications suggests several potential issues such as class imbalance and/or batch effect. The training set was almost perfectly balanced but the validation set had a severe class imbalance. This imbalance could lead to the model being overly confident in predicting malignant cases while almost entirely neglecting the non-malignant ones. We took steps to address the class imbalance by performing oversampling, undersampling, class weights, L2 regularization, and adding dropouts on the validation set, but the metrics never improved. Due to non-improvement we believe there is a possibility that a batch effect occurred during the collection or processing of the non-malignant images in the validation set. A batch effect refers to unintended variations in data caused by differences in how data is collected or processed in different batches. If the non-malignant images in the validation set were collected or processed differently than those in the training set, this could have introduced biases that negatively impacted the model's performance on non-malignant cases. Differences in imaging conditions, equipment, or preprocessing steps could all contribute to this effect [10].

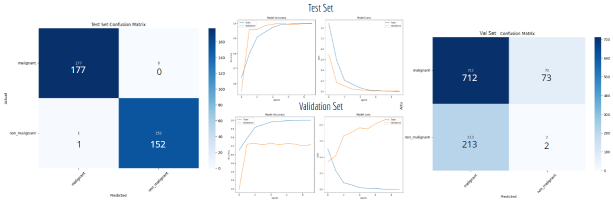

Figure 8: CNN_1 Model Confusion Matrices and Plots for Test and Validation Sets

CNN_2, trained on segmented image arrays, demonstrated strong performance on the test set, with an accuracy of 98.50% and a low loss of 0.04. The precision, recall, and F1-score were consistently high at 98%, indicating the model's ability to effectively classify both malignant and non-malignant cases in the training data.

In the validation phase using the DLCTLungDetectNet dataset, CNN_2 achieved a validation accuracy of 77.70% with a loss of 0.91, showing better generalization compared to CNN_1. However, a closer look at the classification metrics for the validation data reveals significant differences in performance between the two classes.

CNN_2 was highly effective at identifying malignant cases, with a recall of 98%, meaning it correctly identified nearly all malignant cases in the validation set. However, the model struggled with non-malignant cases, achieving only 28% precision and 2% recall, which resulted in an F1-score of 0.04. This suggests that while the model is very good at detecting malignancy, it is not reliable in recognizing non-malignant cases. Similar to CNN_1, CNN_2's strong performance in classifying malignant cases but poor performance on non-malignant cases suggests potential issues of data imbalance and/or batch effect that could have affected the model's ability to generalize to non-malignant cases.

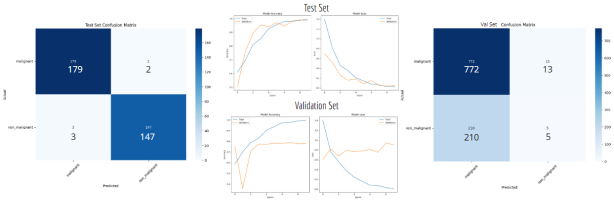

Figure 9: CNN_2 Model Confusion Matrices and Plots for Test and Validation Sets

8 Summary

This project underscores the challenges and opportunities in developing ML tools for lung cancer diagnosis using CT imagery. Our dual approach, leveraging both CNNs for raw and segmented image analysis and texture-based Logistic Regression models, provided valuable insights into the strengths and limitations of each method. The CNN, while highly effective during testing, exhibited a significant drop in performance during external validation, likely due to overfitting and/or potential batch effects within the validation dataset. On the other hand, the texture-based Logistic Regression model offered more consistent performance across datasets. This suggests that traditional ML techniques, when combined with robust feature extraction methods like GLCM, can yield reliable and interpretable results, especially in the context of external validation. The successful application of the UNET-based segmentation model further demonstrated the importance of image preprocessing in enhancing the accuracy of predictive models.

In conclusion, our findings advocate for a hybrid approach that balances the power of deep learning with the interpretability and stability of traditional methods. Future work should focus on refining segmentation techniques, addressing class imbalances, and exploring additional feature extraction methodologies to further improve model generalizability for broader application. Our toolset provides a foundation for continued research and development in this critical area of medical diagnostics.

References

[1] Al-Yasriy, Hamdalla, and Muayed Al-Huseiny. "The IQ-OTH/NCCD lung cancer dataset." Kaggle, Mendeley Data, 2023. Retrieved from url Accessed 27 August 2024.

[2] Dharpure, Harshal. "DLCTlUNGDetectNet - Lung Tumor Dataset." Kaggle, Kaggle, 2024. Retrieved from url Accessed 27 August 2024.

[3] Eichkitz, Christoph , Schreilechner, M.G., de Groot, Paul, Amtmann, Johannes. (2015). Mapping directional variations in seismic character using gray-level co-occurrence matrix-based attributes. Interpretation. 3. T13-T23. 10.1190/INT-2014-0099.1.

[4] Falk, Thorsten, et al. "U-NET: Deep Learning for Cell Counting, Detection, and Morpometry." Research Gate, Nature Methods, January 2019. Retrieved from url. Accessed 17 August 2024.

[5] Ge, Yanqiu, et al. "Predicting post-stroke pneumonia using deep learning neural network approaches." Pub Med, Int J Med Informatics, December 2019, url. Accessed 17 August 2024.

[6] Gobara, Mohamed. "Lung Cancer 98.8% Custom CNN Model." Kaggle, Kaggle, January 2024. url . Accessed 01 August 2024.

[7] Naiborhu, Josua. "How to Interpret Black Box Models using LIME (Local Interpretable Model-Agnostic Explanations)." freeCodeCamp, 17 October 2022. url . Accessed 28 August 2024.

[8] National Cancer Institute. "Cancer of Any Site — Cancer Stat Facts." SEER Cancer, National Cancer Institute, 2024 Retrieved from url Accessed 27 August 2024.

[9] Shafi, Imran, et al. "An Effective Method for Lung Cancer Diagnosis from CT Scan Using Deep Learning-Based Support Vector Network." NCBI, Cancers, 6 November 2022. url. Accessed 28 August 2024.

[10] Sprang, Maximilian, et al. "Batch effect detection and correction in RNA-seq data using machine-learning-based automated assessment of quality - BMC Bioinformatics." BMC Bioinformatics, 14 July 2022. Retrieved from url Accessed 19 August 2024.

[11] Wright, George W., et al. "A Probabilistic Classification Tool for Genetic Subtypes of Diffuse Large B Cell Lymphoma with Therapeutic Implications." Cancer Cell, vol. 37, no. 4, 2020, pp. 551-568. Cancer Cell. Retrieved from url Accessed 27 August 2024.