Malignancy-Based Classification of CT Scan Images for Lung Cancer Patients

A Deep Learning Approach for Malignancy Prediction

(The Assignment's Report of Applied AI in Biomedicine Course)

Christian Ferrareis[1], Bahram Hedayati[2], and Kristina Tas[3]

[1]Student of Computer Science and Engineering, christian.ferrareis@mail.polimi.it, 10725804

[2]Student of Telecommunication Engineering, bahram.hedayati@mail.polimi.it, 10870276

[3]Student of Biomedical Engineering, kristina.tas@mail.polimi.it, 10968804

1 Introduction

Cancer is a major social, public health, and economic problem in the 21st century, responsible for almost one in six deaths (16.8%) and one in four deaths (22. 8%) from non-communicable diseases (NCDs) worldwide [1]. The late-stage diagnosis of lung cancer significantly affects its prognosis, contributing to a low 5-year survival rate of less than 20% in most cases [2]. Traditional cancer detection relies on radiologists analyzing CT scans, which can be time-consuming and prone to human error. Early-stage lung cancer is difficult to detect because small tumors often look like benign nodules which are small masses of tissue that may indicate malignancy. Implementing lung cancer screening programs has been a crucial step toward early diagnosis and intervention. However, the effectiveness of such programs heavily relies on the accurate detection and classification of nodules [3]. Automated computer-aided detection (CAD) systems and deep learning-based approaches have shown great potential in assisting radiologists in identifying malignant nodules with high accuracy as they can detect subtle abnormalities invisible to the human eye [4].

This study proposes a deep learning-based malignancy classification model to enhance the detection of lung nodules in CT scan images. The goal is to improve early diagnosis accuracy, thereby increasing the chances of effective treatment and reducing lung cancer mortality rates. To do this, we did some standardization tasks, performed a number of pre-processing techniques, and implemented four different classifiers based on the images' type and the sort of classification including multi-class and binary classifications.

2 Materials and Methods

2.1 Machine Learning Framework

In order to achieve the goal mentioned in the introduction, we followed a standard workflow implemented in [5] and modeled four different classifiers based on the images' type and the sort of classification including multi-class and binary classifications as we were asked to do in the assignment. Figure 1 shows the workflow of the proposed framework for the detection of malignancy scores with respect to each classifier.

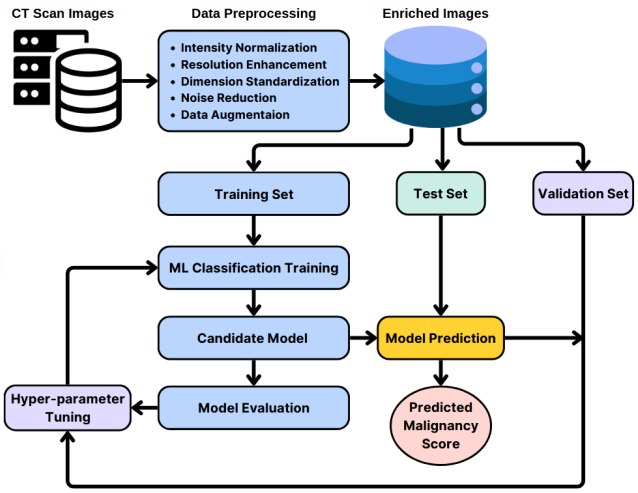

Figure 1: Work flow diagram of ML-based framework for detection of malignancy score of CT scan images

2.2 Dataset Exploration

Data exploration is crucial for understanding patterns, detecting anomalies, and ensuring data quality before analysis or modeling. The given dataset includes two categories of CT scan images per patient: full view and nodule view. Each view consists of 2363 slices of a CT scan. In general,

the output of a performed CT scan includes many slices covering a volume. The slices in the dataset are those with the largest nodule area. Each image is assigned with a specific level of malignancy falling between 1 and 5. The dataset is highly unbalanced in terms of malignancy score for both multiple and binary classes as you can see in Figure 2. This imbalance can introduce bias to the Artificial Intelligence (AI) model, as overrepresentation of one class and underrepresentation of others may cause the model to favor the majority class, leading to inaccurate predictions. Ensuring balanced datasets is crucial to avoid misdiagnosis, ineffective treatments, and healthcare disparities [6]. Therefore, it is necessary to exploit the methods in order to balance the dataset before training the model.

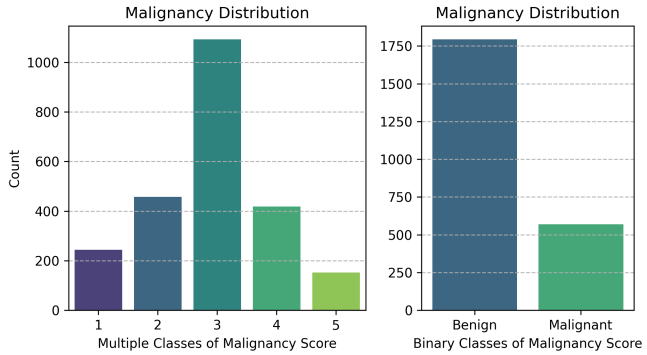

Figure 2: Unbalanced distribution of class (malignancy score)

When it comes to the brightness, the shape of the lungs, and the contrast of images, we observed that they are quite different as some samples are illustrated in Figures 3 and 4. Furthermore, the shapes of nodule slice images vary from (44×45) to (124×108) which imposes size standardization as a pre-processing step. The presence of random noise, haze-like effect, and darkness are the other issues regarding the quality of images that should be addressed effectively.
Additionally, the slices vary significantly, likely due to differences in imaging machines and their positions within the CT scan.

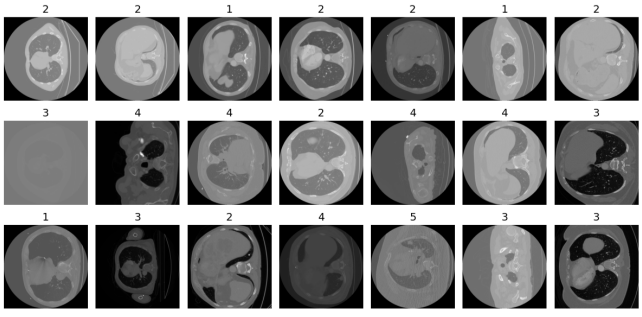

Figure 3: Some samples of full slice images. The label of each image is specified on the top.

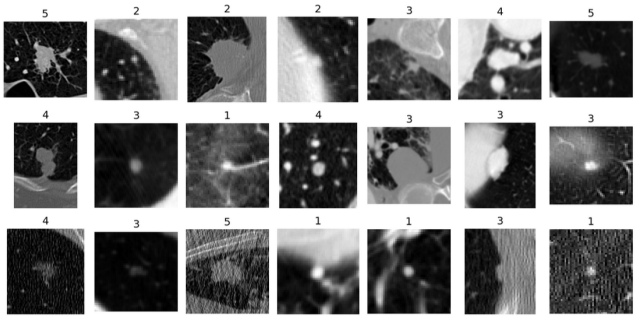

Figure 4: Some samples of nodule slice images. The label of each image is specified on the top.

In order to tackle the aforementioned issues with the unbalanced class distribution, we explored different techniques and tried to address the class unbalancedness before training the models. Furthermore, we used a variety of image processing methods to enhance image quality and improve learning. Details of these methods are provided in the Data Pre-processing section.

2.3 Data Pre-processing

Raw healthcare data is often noisy, incomplete, and inconsistent. Preprocessing ensures that the dataset is clean, balanced, and standardized before feeding it into AI models [6].

CT scans data contains a large amount of irrelevant information for the purpose of this task. Therefore, we hypothesized that filtering image pixels based on Hounsfield units would be important in reducing the input data for the model. In order to ensure consistency across the samples in the dataset, we employed techniques such as intensity normalization, contrast enhancement, dimension standardization, noise reduction, lung masking and background removal.

To further refine the input data, we experimented with a lung segmentation technique to filter out pixels that do not correspond to lung tissue.
In the following sections, we detail how we carried out these data preprocessing steps.

2.3.1 Intensity Filtering and Normalization

The file format of the images in the dataset is the NRRD (Nearly Raw Raster Data). It is a flexible and efficient format for storing n-dimensional raster data, commonly used in medical imaging and scientific visualization to store CT, MRI, and other volumetric data along with metadata in a separate or embedded header [7]. The values in each NRRD file are scaled in the Hounsfield scale (HU), which is a quantitative scale used in CT imaging to measure tissue density. On the HU scale, air is -1000 HU, water is 0 HU, and dense bone is around +1000 HU, helping to differentiate tissues based on their

X-ray attenuation properties [8]. Since the variety of intensities is extremely high, we clipped the intensities to remove unnecessary intensities for analyzing the lung and related tissues. Following this, we normalized the values to the range [0,1] to standardize the input for the model.

2.3.2 Contrast Enhancement

After the intensity filtering and normalization step, we performed the Gamma transformation, which is a nonlinear transformation used in image processing to adjust the brightness and enhance the contrast of an image [9]. The optimal Gamma value found for the majority of images is 1.35 which improves the quality of images while preserving the details.

In addition to Gamma transformation, we performed CLAHE (Contrast Limited Adaptive Histogram Equalization) which is an image enhancement technique that improves contrast by applying local histogram equalization to small regions of an image instead of the whole image, preventing over-enhancement and noise amplification. It limits the contrast in each region to avoid excessive brightness differences, making it useful for medical images, low-light conditions, and X-rays. This method ensures balanced contrast enhancement while preserving important details [10]. Fine-tuning CLAHE's parameters is challenging and can be specified after doing a considerable number of experiments. However, we could find clipLimit=2.5 and tileGridSize = (8, 8) after studying some papers like [11] and [12]. You can see the results of performing these operations on some of the full-slice images in Figure 5.

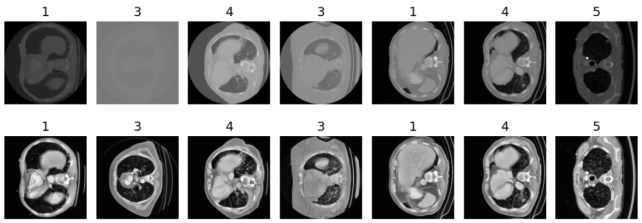

Figure 5: Some of pre-processed full slice images. The bottom images are preprocessed editions of the upper ones after intensity standardization and performing Gamma transformation.

2.3.3 Dimension Standardization

The dimension of full-view images is fixed (512×512), whereas nodule-view images have different dimensions. Thus, nodule slice images need resizing as they are too much different in terms of dimension. Figure 6 demonstrates a general view of the dimension distribution of nodule slices; this is a simplified view that shows only shapes present more than 25 times.

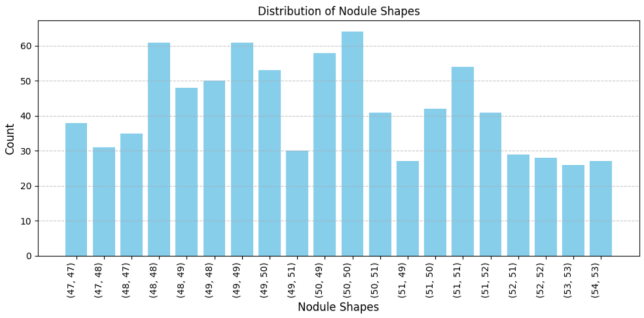

Figure 6: Shape distribution of nodule slice images. For limited space reasons, only shapes that have a frequency greater than 25 are shown.

We consider a mean of the dimension majority as the standard dimension for nodule slices. Some samples of the resulting images of nodule slices after the pre-processing phase are shown in Figure 7.

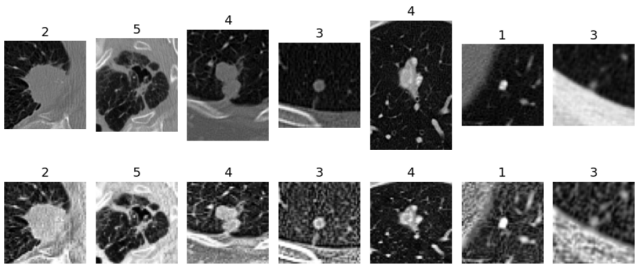

Figure 7: Some of pre-processed nodule slice images. The bottom images are preprocessed editions of the upper ones after intensity standardization, resizing, and performing Gamma transformation.

2.3.4 Noise Reduction

Detection of tumors accurately necessitates high-quality images with minimal noise interference. Given the limited number of images in the dataset, it's crucial to preserve as many images as possible, filtering out only those with excessive noise that could impede analysis. A practical approach involves computing the Laplacian variance of each image to quantify the level of detail and noise. By analyzing the distribution of these variances across the dataset as shown in Figures 8, 9, and 10 we can clearly see that some images contain extremely high Laplacian Variance which can be indicated as super-noisy images. Thus, we can establish a threshold to identify and exclude only the most noisy images and preserve the majority of the data for tumor detection tasks. This method aligns with the techniques discussed in [13].

Specifying a certain threshold for excluding noisy images from the dataset can be done by doing a statistical analysis of the Laplacian variance values to figure out how many slices will be removed with

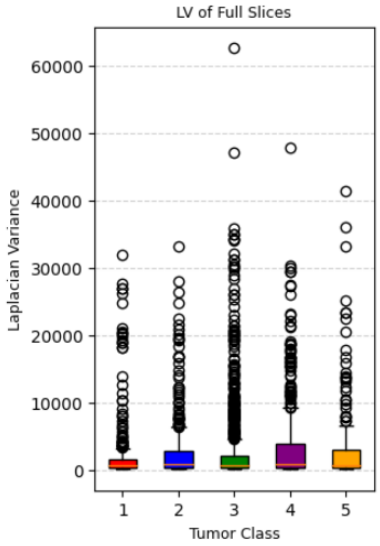

Figure 8: Laplacian Variance values of full-slice images w.r.t. the tumor class

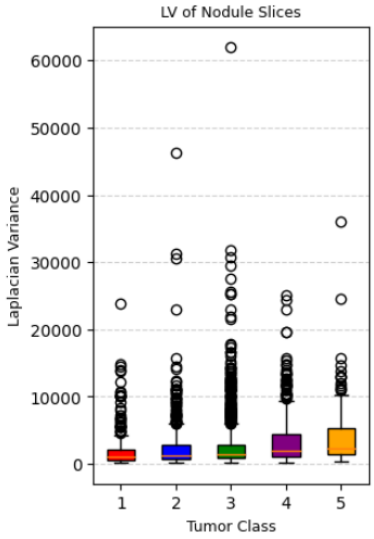

Figure 9: Laplacian Variance values of nodule-slice images w.r.t. the tumor class

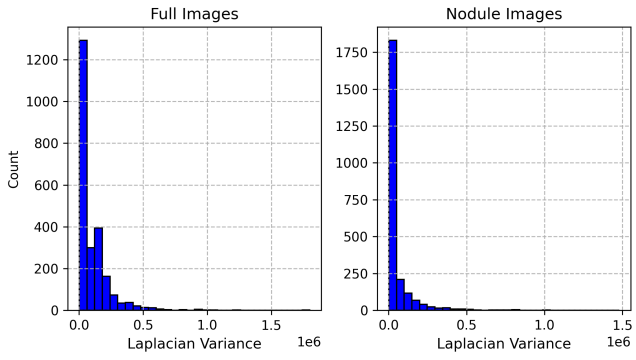

Figure 10: Laplacian Variance of Full and Nodule Images.

respect to the percentile of the Laplacian variance as illustrated in Table 1.

Furthermore, doing some experiments to train the model and check the performance metrics can be helpful for making an optimal decision on noisy

Percentile	Noisy Full Slices	Noisy Nodule Slices
99	23	24
98	48	48
97	71	71
96	95	95
95	119	119
94	142	142
93	166	166
92	189	189
91	213	213
90	237	237

Table 1: The number of noisy images with respect to the percentile of the Laplacian variance values.

image removal procedure. Looking at Table 1, it is clearly seen that there is a linear correlation between the percentile of the Laplacian variance distribution and the number of noisy images for both full slices and nodule slices. According to the fact that we do not want to exclude so many images from the dataset and retain as many images as possible, we decided to exclude the 2% most noisy slices, which equals 71 images from full and nodule slices each. You can see some examples of the most noisy images that exist in the dataset even after contrast enhancement in Figure 11.

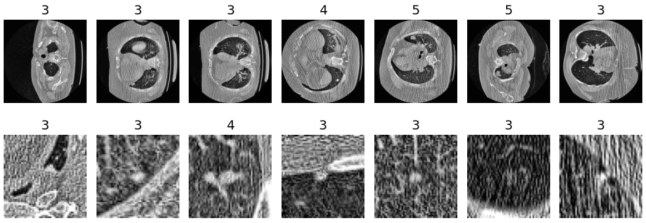

Figure 11: Example of noisy full images (first row) and noisy nodule images (second row)

2.3.5 Lung Masking and Background removal

CT scan images contain a vast amount of information, much of which is not relevant to the specific task of lung analysis. To enhance model performance and focus on lung-related structures, we applied a lung masking technique to isolate the lung regions while filtering out extraneous elements such as bones, soft tissue, and air pockets.

This step ensures that only lung-relevant features contribute to the learning process, reducing noise and improving the focus on key anatomical structures. To achieve this we employed the Watershed algorithm, a classical image processing technique commonly used for separating different objects in an image.

The Watershed algorithm treats a grayscale image

as a topographic surface, where pixel intensity represents elevation. Using user-defined markers, the algorithm floods the regions from local minima until adjacent basins merge at watershed lines. This allows for the extraction of lung boundaries with high accuracy. The segmented lungs were then overlaid on the original CT images, with non-lung pixels set to zero. Figure 12 shows some of the resulting images.

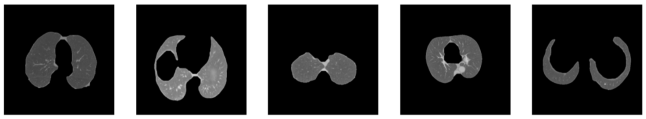

Figure 12: Example of lung-segmented images

2.4 Addressing Class Imbalance

As mentioned before, the dataset is highly imbalanced, meaning some classes have far more images than others. This imbalance can cause the deep learning model to favor majority classes, leading to poor performance in minority classes. We experimented several approaches to handle the class imbalance problem, as outlined below.

2.4.1 Class Upsampling and Downsampling

2.4.2 Class Weighting

An alternative approach to the just mentioned method is the class weighting approach, which assigns higher weights to underrepresented classes and lower weights to overrepresented ones. The weights are computed for multi-label and binary classifications separately using the formula (1).

$$w_i = \frac{n_{\text{samples}}}{n_{\text{classes}} \times n_i} \tag{1}$$

where:

- w_i: The weight for class i
- $n_{samples}$: The total number of images in the dataset
- $n_{classes}$: The total number of classes in the dataset
- n_i: The number of samples with class i in the dataset

The corresponding computed weights for each class using the scikit-learn library are shown in Table 1. The class weighting approach helps the model penalize misclassifications of minority classes more heavily.

Class	Weight
1	1.929
2	1.053
3	0.426
4	1.149
5	3.176
Benign	0.655
Malignant	2.109

Table 2: Weights of different classes of the dataset

2.4.3 SMOTE

Since the implementation of the previously discusses methods did not help in the improvement of the model, we tried two alternative approaches consisting in rebalancing the dataset with synthetically generated training samples. The first technique we mention is Synthetic Minority Over-sampling Technique (SMOTE), a widely used method for addressing class imbalance in machine learning by generating synthetic samples for the minority class. Instead of simply duplicating existing samples, SMOTE creates new instances by interpolating between real data points. This is done by selecting a minority class sample, identifying its k-nearest neighbors, and generating a new sample along the line segment between the original point and one of its neighbors. By enriching the minority class in this way, SMOTE helps improve model performance by reducing bias and enhancing generalization. Although originally intended for tabular data, we adapted it for the use on our images by flattening the channel dimension, and reshaping the image back to 3 dimensions after the performed operation. Considering the underlying mechanism behind SMOTE, there's only a limited number of quality samples it can generate. Therefore, we used it to duplicate the number of minority class samples, at best. Transfer learning is particularly beneficial in scenarios with limited labeled data [16], such as the given dataset. Below is a brief description of the main backbone families used.

2.4.4 Synthetic Sample Generation with GAN

As a last approach to tackle the class imbalance issue, we experimented Generative Adversarial Networks (GANs) to generate new, synthetic, training samples. GANs are a class of DL models that belong to the unsupervised learning family. They are primarily used for generating synthetic data when there is insufficient real data. In this project they were employed for both rebalancing the dataset and to increase the number of training samples among all the classes, to deal with the data

scarcity issue.
A GAN consists of two neural networks Generator and Discriminator that compete against each other in a game-theoretic framework. The goal of the Generator is to fool the Discriminator, while the goal of the Discriminator is to correctly distinguish between real and fake samples. The training procedure involves G attempting to maximize the probability of D making a mistake, leading to a minimax two-player game [17]. In each iteration, if the Discriminator correctly detects fake data, the Generator is penalized. If the Generator successfully fools the Discriminator, it is rewarded. Over multiple iterations, the Generator improves and starts producing highly realistic data.

Conditional GAN (cGAN) is an extension of the traditional GAN framework that incorporates additional information into both the generator and discriminator models. This conditioning information can be in the form of class labels, attributes, or any other data that provides context, enabling the generation of outputs with specific desired characteristics. By integrating this auxiliary data, cGANs offer enhanced control over the data generation process, allowing for the creation of more targeted and relevant samples [18].

Synthetic Data Generation with cGAN

As said before, to address dataset imbalance and expand the training set, we employed a cGAN model. Given a vector of random noise concatenated with the 1-hot encoding of the class, the cGAN outputs a synthetic sample for the given class. First, it was used to generate additional samples for underrepresented classes, ensuring a more balanced dataset and improving the model's ability to learn from all categories. Additionally, the cGAN helped expand the training set by producing diverse synthetic images, reducing the risk of overfitting and enhancing generalization. This approach aimed to provide a more representative dataset, ultimately improving classification performance. In this assignment, a cGAN was exploited to upsample each class to 1330 samples, resulting in 6650 samples in total.

2.5 Methodology: DL Models

Deep Learning (DL) is a subset of machine learning (ML) and AI that extracts a complex hierarchy of features from images by its self-learning ability. It involves neural networks with many layers that extract a hierarchy of features from raw input images. Several types of DL approaches have been developed for different purposes, such as object detection and segmentation in images [21]. In this section we discuss some of these DL models we found useful and effective for the given dataset in terms of malignancy score prediction.

Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are the algorithms most commonly applied to images. CNN architectures are increasingly complex, with some systems having more than 100 layers, which means millions of weights and billions of connections among neurons. A typical CNN architecture contains multiple convolution, pooling, and activation layers [22].

The output scores from the final CNN layer are then passed to a multi-layer perceptron (MLP) classifier, which processes these features to generate classification scores. Finally, a softmax activation function is applied to normalize these scores into a probability distribution over the possible labels, ensuring that the model produces interpretable predictions. During training, an optimizer is used to update the model's weights based on the gradients computed through backpropagation, gradually refining them until they converge to an optimal or stable state. A simple schema of a CNN is illustrated in Figure 12.

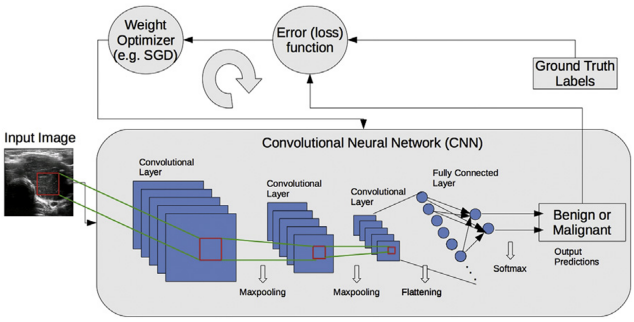

Figure 13: A schematic illustration of a CNN in a DL process

We implemented this structure as a baseline model to develop more robust models. Specifically, we leveraged Transfer Learning by experimenting with different pre-trained backbones as feature extractors. This approach was chosen based on the hypothesis that training feature extractors from scratch would be impractical due to the limited size of the given dataset.

2.5.1 Model Architecture and Transfer Learning

Transfer learning is a technique in ML where a model developed for a particular task is reused as a starting point for a model on a different but related task. This approach is especially beneficial when dealing with limited data in the target domain, like the given dataset in this assignment, as it allows

leveraging knowledge from a source domain where ample data is available. VGG, MobileNet, EfficientNet, and ResNet families contain pre-trained models on large datasets such as ImageNet to capture a wide range of features that can be fine-tuned for specific tasks. The goal is to improve performance and reduce training time, by making use of good visual features extracted by these pre-trained backbones. Utilizing these pre-trained models and fine-tuning them can significantly enhance image analysis and classification tasks in medical images. The following provides an overview of the backbone families used in this project.

VGG

VGG is characterized by its simplicity and depth, utilizing a reduced number of layers (whether 16 or 19, depending on the specific model) composed of small convolutional filters. This design choice allows the network to capture intricate features while maintaining manageable computational complexity [23]. As we learned during the practical lessons, its architecture's uniformity and depth have made it a benchmark in image recognition tasks.

MobileNet

The MobileNet model [24] family consists of lightweight convolutional neural networks designed for efficient performance on mobile and edge devices. They use depthwise separable convolutions to reduce computational cost while maintaining accuracy. MobileNet models come in different versions (V1, V2, and V3), each improving efficiency and accuracy. They are commonly used for transfer learning in resource-constrained environments.

ResNet

ResNet [25] addresses the gradient degradation problem (known as Vanishing Gradient) in deep networks by incorporating residual learning. This is achieved through shortcut connections that enable the network to learn identity mappings, facilitating the training of much deeper networks without drops in performance [15]. We saw a simple architecture of ResNet during the practical lessons which is shown in Figure 13.

EfficientNet

EfficientNet [26] introduces a compound scaling method that uniformly scales network depth, width, and resolution using a set of fixed scaling coefficients. This approach leads to a family of models that achieve state-of-the-art accuracy while being computationally efficient. EfficientNet-B0, the baseline model, serves as the foundation for this scalable architecture [15]. These architectures are often employed in transfer learning scenarios.

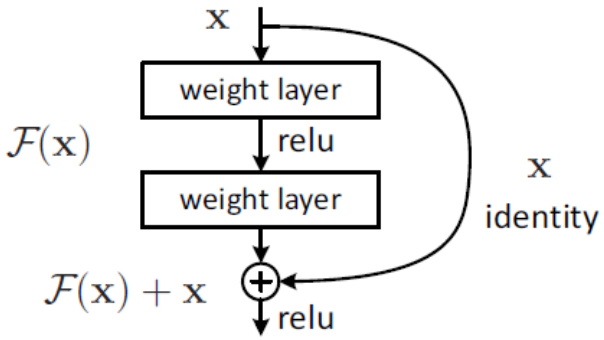

Figure 14: The ResNet architecture

2.5.2 Feature Classifier

Since the pretrained backbones serve as good feature extractors, they output high-level representations of the input images rather than direct classification results. To effectively leverage these representations, different classification strategies were explored.

Given the limited dataset size, the goal was to find a balance between model complexity and generalization capability. In particular, two classification approaches were considered: Multi-Layer Perceptron (MLP) and Support Vector Machines (SVMs), specifically chosen as an alternative strategy to address the challenges of the limited dataset size.

Multi-Layer Perceptron (MLP)

MLP [27] is a fully connected neural network that processes extracted features through multiple dense layers in a hierarchical manner, progressively reducing the input size while learning to effectively utilize the features for the final classification task. We experimented with different hyperparameters, including the number of layers and units, activation functions, and regularization techniques, to optimize performance and prevent overfitting.

Support Vector Machines (SVMs)

SVM [28] is a classical machine learning approach that aims to find the optimal hyperplane that best separates different classes in the feature space. They are particularly effective in high-dimensional spaces and work well with small datasets by maximizing the margin between classes.

We experimented with different hyperparameters, such as the kernel function (used to project the points in a higher-dimensional space) and regularization parameters.

2.6 Model Training

The dataset was split so that 80% of the available data was used for training, while 20% was used for validation. This split allows the model to learn the majority of the data while setting aside a portion for hyperparameter tuning. The test set will be provided to us after the submission of this report, as per assignment instructions.

2.6.1 Data Augmentation

Accurate classification of medical images using deep learning models often necessitates a substantial number of diverse training samples to achieve high accuracy. To enhance model generalization, we applied data augmentation, as the dataset's number of samples per class were insufficient to capture underlying patterns. Data augmentation refers to a group of techniques whose goal is to battle a limited amount of available data to improve model generalization and push sample distribution toward the true distribution [14]. There are different augmentation strategies and performing a certain combination of them depends on the dataset. Data augmentation techniques that we used in this assignment are:

- *Rotation:* This involves rotating images by a certain degree, aiding models in recognizing objects from various orientations. We used a rotation range of 30 degrees
- *Flipping:* Horizontal and vertical flips create mirror images, enhancing the model's ability to generalize across different spatial orientations.
- *Zooming:* It involves scaling images in or out, allowing models to become invariant to size changes.
- *Shearing:* It skews the image along the x or y-axis, providing a perspective shift that helps models learn from distorted versions of the original images.

We experimented with both online and offline data augmentation. Online augmentation was performed using the ImageDataGenerator, a Keras object which applies transformations on-the-fly and provides augmented samples in batches during training. Offline augmentation, on the other hand, was applied during the pre-processing stage, expanding the dataset with additional samples generated through random transformations.
For all classifiers, for both online and offline approaches, we used a rotation range of 30, a shear range of 0.2, a zoom range of 0.2, and applied only horizontal and vertical flipping.

2.6.2 One-hot Encoding

One-hot encoding is a technique used to convert categorical data into a numerical format suitable for deep learning models such as CNNs, which require categorical labels in a structured numerical format for effective training. In this method, each class is represented as a binary vector, where only one element is "1" (indicating the presence of that class), and all other elements are "0". One-hot encoding ensures that each category is independent, preventing unintended hierarchical relationships and avoiding wrong weight adjustments in DL models. Therefore we applied the one-hot encoding on the labels before training each classifier.

2.6.3 Model Specifications

In the case of all tested models, we loaded their pre-trained versions and topped them with a convolutional layer, max pooling layer, flatten layer, dropout layer, 3 dense layers, and another dropout layer, followed by the final dense layer with the number of output neurons and activation function depending on whether it was used for binary or multi-class classification. The hyper-parameters used during the modeling were defined as follows:

- *Activation Function:* ReLU
- *Optimizer:* Adam
- *Weight Decay:* 0.0001
- *L1 regularization:* Applied with a weight of 0.00001 to the loss function
- *Loss Function:* Binary or Categorical Crossentropy, depending on the task at hand
- *Learning Rate:* 0.0001
- *Number of Epochs:* 50
- *Batch Size:* 64

To prevent overfitting, we implemented early stopping and a learning rate scheduler as part of the training strategy. *Early Stopping* monitors the validation loss and stops training when no improvement is observed for 10 epochs, restoring the best weights to ensure optimal model performance. Additionally, we used a *ReduceLROnPlateau* scheduler to dynamically reduces the learning rate when the model's validation performance stops improving for a predefined number of epochs, in this case 5 (patience parameter). When validation loss stagnates, the learning rate is reduced by 0.3, ensuring

smoother convergence and preventing the model from getting stuck in local minima.

3 Experimental Results

3.1 Performance Metrics

The effectiveness of classifications done by each classifier measured by the performance metrics that are listed below:

- *Accuracy*: Represents the proportion of correctly classified instances out of the total instances evaluated. While accuracy offers a general overview, it may not fully capture the model's performance in cases of class imbalance.
- *Precision*: Indicates the proportion of true positive predictions among all positive predictions made by the model.
- *Recall*: Measures the proportion of actual positive cases that the model correctly identifies.
- *F1 Score*: The harmonic mean of precision and recall and useful for situations that there is an uneven class distribution.
- *AUC-ROC*: Area Under the Receiver Operating Characteristic Curve is the plot of the true positive rate against the false positive rate at each threshold setting. AUC is a number that summarizes the ROC curve. It represents the probability that the model ranks a randomly chosen positive sample higher than a randomly chosen negative sample.

3.2 Results of Full-slice Classifiers

Several DL models, such as VGG19, MobileNetV2, EfficientNetB0, ResNet50, performed on the full-slice images. In Table 3 are the results of multi-class full-slice classifiers for the two most successful architectures. For the best-performing model, EfficientNetB0, we report further results. The development of loss values throughout the training is showcased in Figure 15, while the development of accuracy is in Figure 16. Although both loss and accuracy vary, the implementation of early stopping prevented overfitting from occurring. Lastly, the confusion matrix is displayed in Figure 17, showing high specificity, and the ROC curve in Figure 18, further confirming the unsatisfactory performance of the model.

Model	Training		Validation	
	Loss	Accuracy	Loss	Accuracy
EfficientNetB0	1.5281	0.2952	1.5396	0.2896
MobileNetV2	1.4546	0.2762	1.5644	0.2072

Table 3: Training and validation loss/accuracy for the best classification models

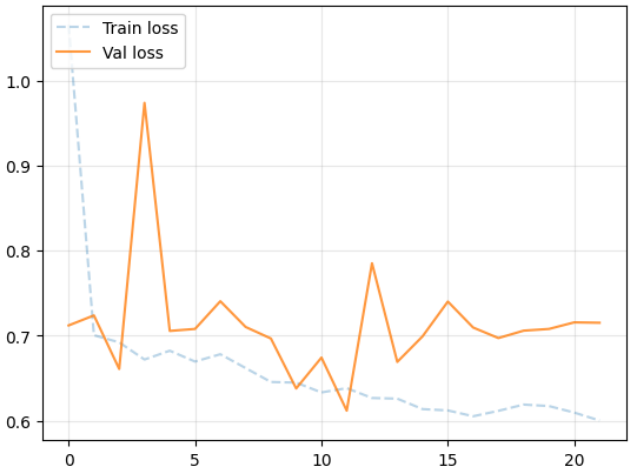

Figure 15: Binary cross entropy for full-slice images.

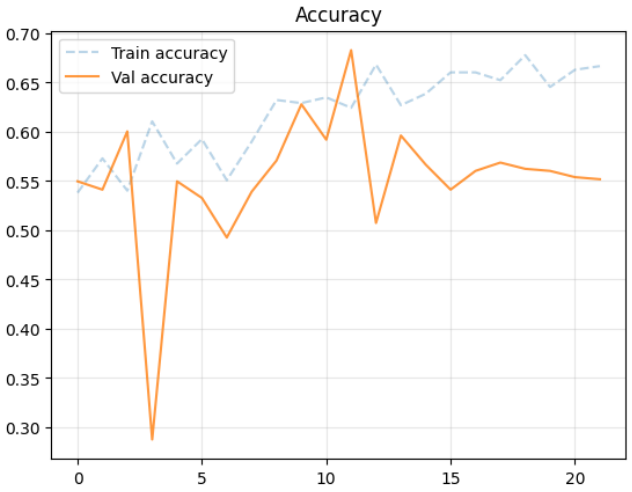

Figure 16: Accuracy of binary classification for full-slice images.

3.3 Results of Nodule-slice Classifiers

Several DL models, such as VGG19, MobileNetV2, efficientNetB0, ResNet50, performed on the nodule-slice images. In Table 4 are the results of multi-class nodule classifiers for the two most successful architectures. For the best-performing model, EfficientNetB0, we report further results. The development of loss values throughout the training is showcased in Figure 20, while the development of accuracy is in Figure 21. Although both loss and accuracy vary, the implementation of early stopping prevented overfitting from occurring. Lastly, the confusion matrix is displayed in Figure 22, showing high specificity, and the ROC

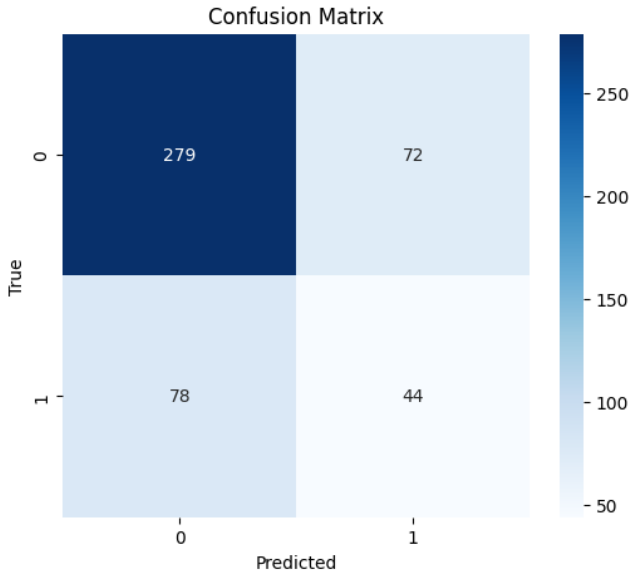

Figure 17: Confusion Matrix of binary classification for full-slice images.

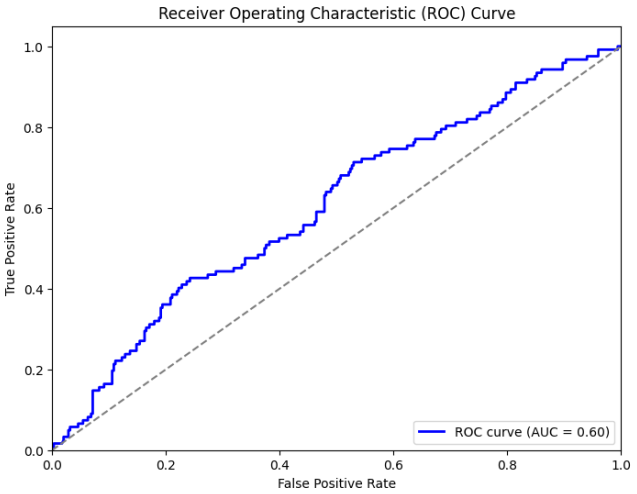

Figure 18: ROC curve of binary classification for nodule-slice images.

curve in Figure 23, further confirming the unsatisfactory performance of the model.

Model	Training		Validation	
	Loss	Accuracy	Loss	Accuracy
EfficientNetB0	1.2950	0.4053	1.4138	0.3404

Table 4: Training and validation loss/accuracy for the best binary classification model

3.4 Explainable AI

Deep learning models, often function as a black box, meaning the decision-making process in these models is unclear. Explainable AI (XAI) aims to open this black box and make AI more trustworthy for healthcare professionals. Therefore, these methods are crucial for interpreting ML models, especially in the medical domain where transparency

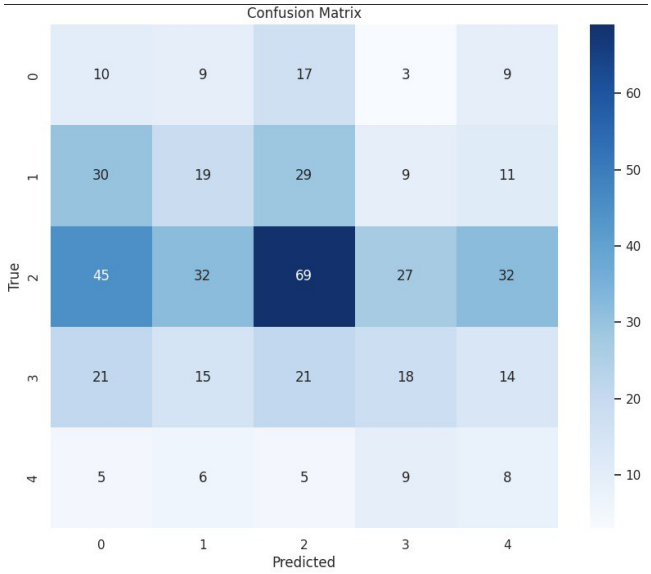

Figure 19: Confusion Matrix of multi-class classification for full-slice images.

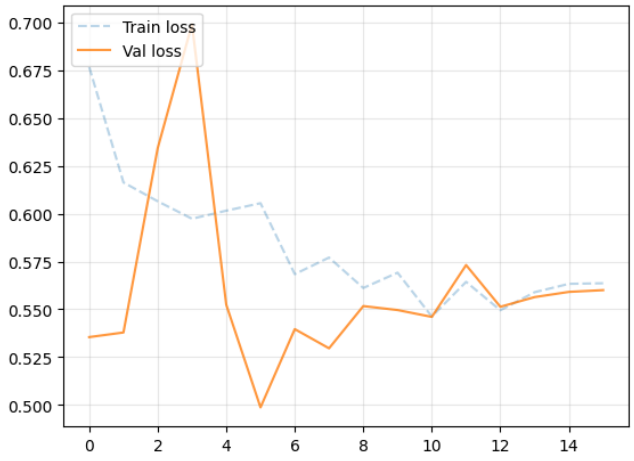

Figure 20: Binary cross entropy for nodule-slice images.

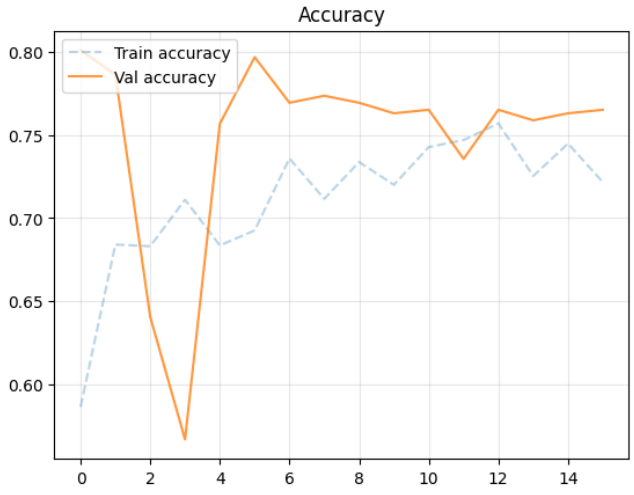

Figure 21: Accuracy of binary classification for nodule-slice images.

and trust are essential. Errors caused by these systems, such as incorrect diagnoses or treatments,

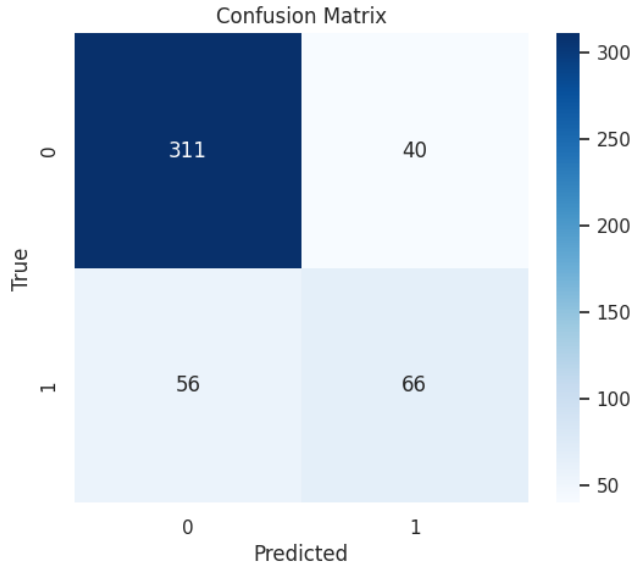

Figure 22: Confusion Matrix of binary classification for nodule-slice images.

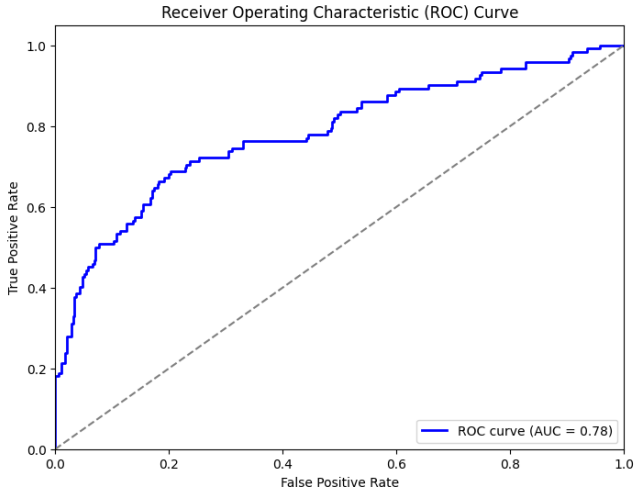

Figure 23: ROC curve of binary classification for nodule-slice images.

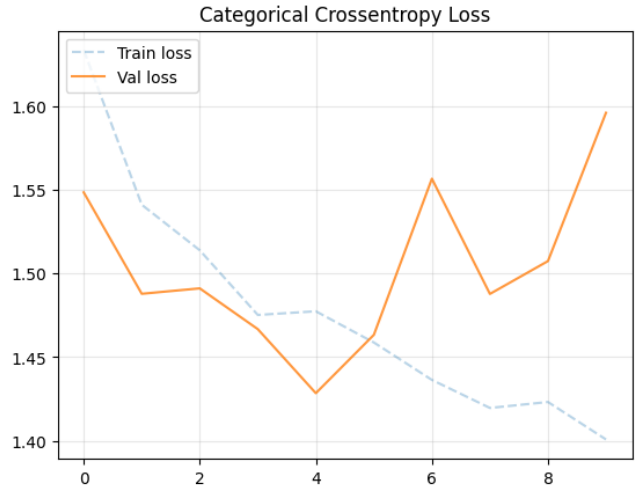

Figure 24: Multi-class cross entropy for nodule-slice images.

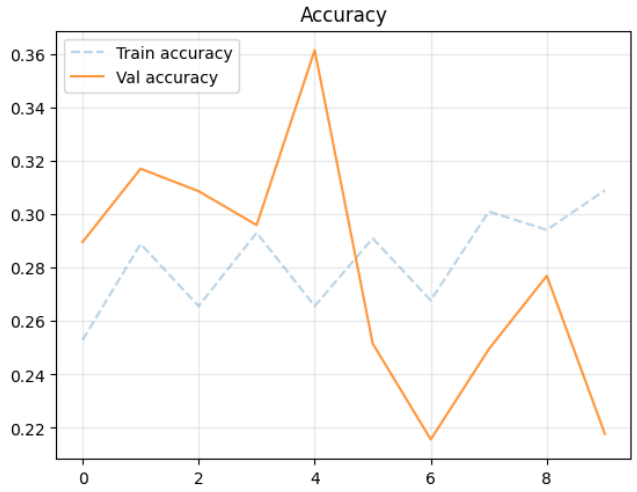

Figure 25: Accuracy of multi-class classification for nodule-slice images.

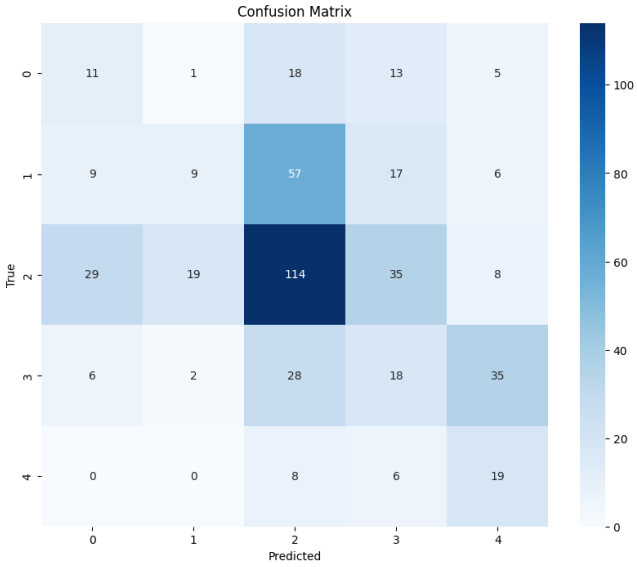

Figure 26: Confusion Matrix of multi-class classification for nodule-slice images.

can have severe and even life-threatening consequences for patients [19]. There are several techniques for making AI models more interpretable. These techniques follow two general paradigms:

- *Model-agnostic approach:* These XAI methods can be applied to any AI model. Local Interpretable Model-Agnostic Explanations (LIME) and Shapley Additive Explanations (SHAP) are two common instances of this approach.
- *Model-Specific approach:* These techniques are designed for particular models and work directly with DL models, like occlusion sensitivity analysis and Grad-CAM.

In this assignment, we exploit the Grad-CAM method to make our model more interpretable as

the main models of this project are deep learning and CNNs specifically.

3.5 Grad-CAM

Gradient-weighted Class Activation Mapping (Grad-CAM) generates a heatmap over an input image, highlighting the most important regions that contributed to the model's prediction. It has been applied to CNNs to enhance the interpretability of tumor classification in CT scans. It can be used to visualize important features, thereby aiding in understanding the model's decision-making process [19]. The working mechanism of Grad-CAM is completely described in [20] and briefly divided into the steps below:

1. *Forward Pass Through the Network:* The input image is fed into the CNN, and activations are obtained from the last convolutional layer. This layer retains spatial information, which is crucial for identifying important regions in the image.

2. *Compute Gradients of the Target Class:* Gradients of the predicted class score are calculated with respect to the activations from the last convolutional layer. These gradients indicate the importance of each neuron in the activations concerning the target class.

3. *Global Average Pooling of Gradients:* The computed gradients are averaged across the spatial dimensions, resulting in weights that reflect the significance of each feature map for the target class.

4. *Weighted Combination of Feature Maps:* The feature maps from the last convolutional layer are combined using the previously calculated weights, producing a coarse localization map that highlights important regions in the image.

5. *Generation of the Heatmap:* An activation function, like ReLU, is applied to the combined map to focus on positive contributions, and the result is upsampled to match the input image dimensions, creating a heatmap that overlays on the original image to visualize the areas influencing the model's decision.

Some resulting images of GRAD-CAM are shown in Figures 23 to 26.

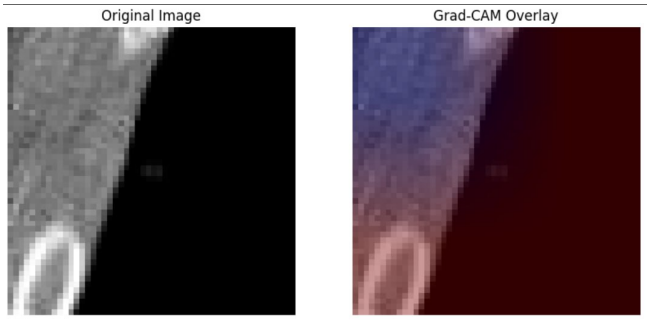

Figure 27: an Original image vs. a Grad-CAM Overlay which specifies effective regions of a nodule slice.

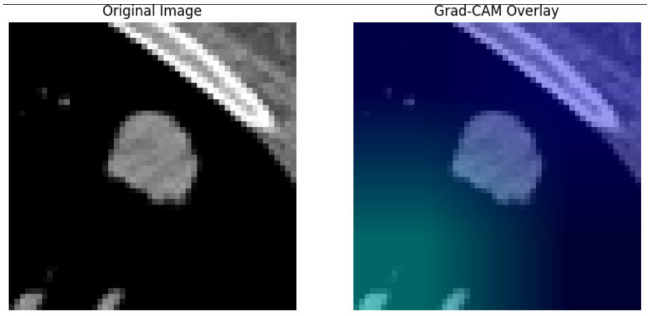

Figure 28: an Original image vs. a Grad-CAM Overlay which could not specify effective regions of a nodule slice.

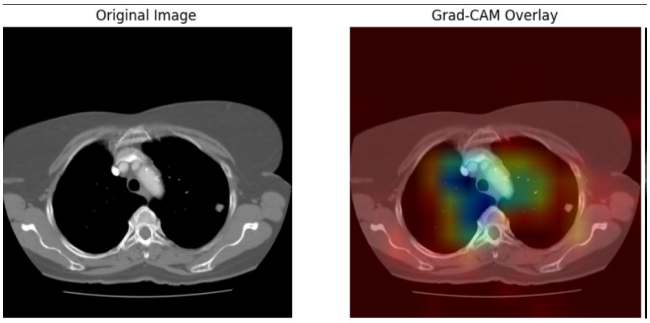

Figure 29: an Original image vs. a Grad-CAM Overlay which specifies effective regions of a full slice.

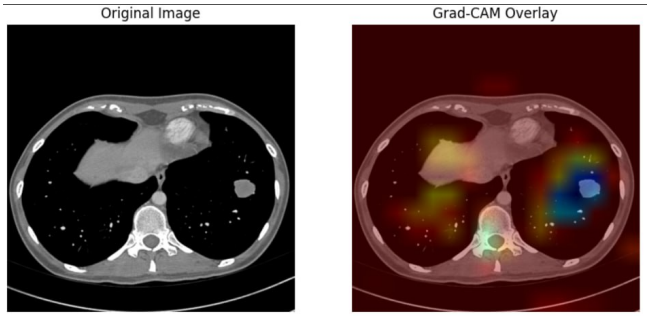

Figure 30: an Original image vs. a Grad-CAM Overlay which could not specify effective regions of a full slice.

4 Discussion and Conclusion

4.1 Limitations

The biggest obstacle we encountered while working on this project was class imbalance. Despite exploring several techniques for tackling it, they did not yield the expected results. For example, cGAN did not provide significant improvements in model performance, possibly due to the complexity of the task and the limitations of the available data. SMOTE also couldn't be used to generate synthetic

data, since a high number of resulting images didn't look realistic. Similarly, background removal was explored in an attempt to eliminate irrelevant regions in the images. However, this process did not enhance the model's ability to focus on the nodules.

Although commonly used to adapt a pre-trained model to a specific task, we didn't implement fine-tuning is. This choice was made considering that unfreezing some of the backbone layers resulted in overfitting, despite the application of regularization techniques. Since we were already experiencing signs of overfitting with the use of transfer learning, further fine-tuning seemed counterproductive and that it would not contribute meaningfully to model performance.

4.2 Full slice images

In full-slice images, the nodules based on which the model should perform classification represent only a few pixels. That being said, these images contain both relevant and irrelevant anatomical structures, with the latter highly influencing classification performance. Classification in multiple classes is inherently more challenging than binary classification, thus, it could've been expected that better results would be obtained in case of binary classification. The fact that the dataset is highly imbalanced only made that issue more evident. The additional anatomical information in full slices ended up introducing noise and making it harder for the model to focus on the specific regions of interest. This is evident from Grad-CAM images (Figures 25 and 26), in which it is clear that the model focuses more on other tissues and backgrounds than the lungs themselves. Binary classification provides better results, as the distinction between the two broader categories is more pronounced than between the original classes. However, the model still struggles with correctly localizing and interpreting the relevant features, meaning that this model cannot be used to help doctors determine the type of cancer at hand.

4.3 Nodule slice images

Nodule images contain only the cancerous nodule and the surrounding tissue, thus removing extraneous structures that the previous models were focusing on. This allows the models to focus solely on the regions of interest, but the class imbalance problem still persists. Multiclassification remains a more complex task as the model failed to capture well the subtle differences between classes. However, the absence of irrelevant anatomical information helped in improving performance compared to full-slice data. Similarly to the full slice case, the binary classifier obtained better results than its multiclass counterpart. Unfortunately, despite implementing measures to tackle class imbalance, the model ended up learning class 0 better. The model exhibits high specificity, meaning it could potentially aid doctors in confirming benign cases.

4.4 Conclusion

In conclusion, the results from classifying full CT slices suggest that improving model performance could be achieved by applying attention mechanisms or performing nodule segmentation as a preprocessing step. By focusing the model's attention on the relevant regions, such as the nodules, these approaches can reduce the impact of irrelevant background information, leading to better feature extraction and improved classification accuracy. Additionally, further exploration of techniques such as advanced loss functions and combining them with already implemented techniques such as data augmentation and synthetic data generation could help address the issue of class imbalance, which remains a challenge in both multiclass and binary classification tasks. Combining these strategies could enhance the model's ability to learn discriminative features, improving both its generalization and its robustness in dealing with imbalanced data.

5 References

[1] Sung, H., Ferlay, J., Siegel, R. L., et al. (2021). Global cancer statistics 2020: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. CA: A Cancer Journal for Clinicians, 71(3), 209-249.

[2] Siegel, R. L., Miller, K. D., & Jemal, A. (2022). Cancer statistics, 2022. CA: A Cancer Journal for Clinicians, 72(1), 7-33.

[3] Aberle, D. R., Adams, A. M., Berg, C. D., et al. (2011). Reduced lung cancer mortality with low-dose computed tomographic screening. New England Journal of Medicine, 365(5), 395-409.

[4] Ardila, D., Kiraly, A. P., Bharadwaj, S., et al. (2019). End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography. Nature Medicine, 25(6), 954-961.

[5] Akkus, Z., Cai, J., Boonrod, A., Zeinoddini, A., Weston, A. D., Philbrick, K. A., & Erickson, B. J. (2019). A survey of deep-learning applications in ultrasound: Artificial intelligence–powered ultrasound for improving clinical workflow. Journal of the American College of Radiology, 16(9), 1318–1328.

[6] Bohr, A., & Memarzadeh, K. (Eds.). (2020).

Artificial Intelligence in Healthcare. Academic Press.
[7] Bourke, P. (2004). NRRD (Nearly Raw Raster Data) File Format Specification. Retrieved from www.paulbourke.net.
[8] Ramphal, R., & Raniga, S. B. (2020). Hounsfield Unit. Published by StatPearls. Retrieved from www.ncbi.nlm.nih.gov.
[9] Gonzalez, R. C., & Woods, R. E. (2008). Digital Image Processing (3rd ed.). Prentice Hall.
[10] Tawfik, N., Emara, H., El-Shafai, W., Soliman, N., Alarni, A. & Abd El-Samie, F. (2024). Enhancing Early Detection of Lung Cancer through Advanced Image Processing Techniques and Deep Learning Architectures for CT Scans. Computers, Materials, and Continua. 81. 271-307.
[11] Fawzi, A., Achuthan, A., & Belaton, B. (2021). Adaptive Clip Limit Tile Size Histogram Equalization for Non-Homogenized Intensity Images. IEEE Access. PP. 1-1.
[12] Khomduean, P., Phuaudomcharoen, P., Boonchu, T. et al. Segmentation of lung lobes and lesions in chest CT for the classification of COVID-19 severity. Sci Rep 13, 20899 (2023).
[13] Ranjbaran A, Hassan AH, Jafarpour M, Ranjbaran B. A Laplacian based image filtering using switching noise detector. Springerplus. 2015 Mar 8;4:119.
[14] Hao R, Namdar K, Liu L, Haider MA, Khalvati F. A Comprehensive Study of Data Augmentation Strategies for Prostate Cancer Detection in Diffusion-Weighted MRI Using Convolutional Neural Networks. J Digit Imaging. 2021 Aug;34(4):862-876.
[15] Yang Y, Zhang L, Du M, Bo J, Liu H, Ren L, Li X, Deen MJ. A comparative analysis of eleven neural networks architectures for small datasets of lung images of COVID-19 patients toward improved clinical decisions. Comput Biol Med. 2021 Dec;139:104887.
[16] Salehi, A. W., Khan, S., Gupta, G., Alabdullah, B. I., Almjally, A., Alsolai, H., Siddiqui, T., & Mellit, A. (2023). A Study of CNN and Transfer Learning in Medical Imaging: Advantages, Challenges, Future Scope. Sustainability, 15(7), 5930.
[17] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). "Generative Adversarial Nets." Advances in Neural Information Processing Systems (NeurIPS).
[18] Coursera Website
[19] Ali S, Akhlaq F, Imran AS, Kastrati Z, Daudpota SM, Moosa M. The enlightening role of explainable artificial intelligence in medical & healthcare domains: A systematic literature review. Comput Biol Med. 2023 Nov;166:107555.
[20] Glass Box Medicine Website
[21] Q. Li and S. Yang, Deep Learning in Object Recognition, Detection, and Segmentation. San Rafael, CA: Morgan & Claypool Publishers, 2018.
[22] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," Neural Computation, vol. 1, no. 4, pp. 541–551, 1989.
[23] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," arXiv preprint arXiv:1409.1556, 2014.
[24] A. G. Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," arXiv preprint arXiv:1704.04861, 2017.
[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.
[26] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," arXiv preprint arXiv:1905.11946, 2019.
[27] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," Nature, vol. 323, no. 6088, pp. 533–536, 1986.
[28] C. Cortes and V. Vapnik, "Support-Vector Networks," Machine Learning, vol. 20, no. 3, pp. 273–297, 1995.