Exploratory Analysis of Cancerous vs. Non-Cancerous Lungs:

A Classification Problem

Scientific Machine Learning Final Project

Reem Fashho
MSE Biomedical Engineering
The University of Texas at Austin
rif252
reem.fashho@utexas.edu

Amanda Nowacki
PhD Biomedical Engineering
The University of Texas at Austin
an29936
anowacki@utexas.edu

Shreya Shukla
MS Information Studies
The University of Texas at Austin
ss223882
shreya.shukla@utexas.edu

Abstract— Cancer is the second leading cause of death worldwide [1], with lung cancer being the second most common type of cancer [2]. Usually, a radiologist will review the scan to determine the diagnosis, but there are cases when the radiologist misses a malignant nodule - in fact, 90% of missed cases are when analyzing a chest CT [4]. To combat this, radiologists have adopted computer-aided diagnosis (CAD) systems that predict whether a nodule on a scan is malignant. The radiologist can then compare their assessment of the scan to that of the CAD system to make a more-informed diagnosis potentially. In this paper we compare the performances of neural network based models, as they classify lung images as cancerous and non-cancerous. For the purpose of validating cancer location, we used the LIME (Local Interpretable Model-Agnostic Explanations) explainability technique to find the regions which led to positive prediction as a cancerous label.

I. Introduction

Lung cancer has surpassed prostate cancer as being the most prominent form of cancer in men and is the second-most prominent form of cancer in women. Lung cancer is typically diagnosed through the analysis of a computed tomography (CT) image (Figure 1). The available CAD systems are designed to be used in complementing the radiologists' diagnoses, not informing them, as lung nodule CAD systems have been known to contain an average of 6.6 false positives per image [5]. Thus, there is a need for these CAD systems to be improved to act as a better "second opinion" to radiologists.

Through this project, we investigate two questions, rather than hypotheses: (1) can the models parse between cancerous and non-cancerous pulmonary CT scans and (2) are the models able to identify the spatial location of cancerous tissue within those scans.

For the first posed question, we turned to machine learning to distinguish between lung tumors and surrounding healthy lung tissue. We evaluated the tumor classification performance of each algorithm through the metrics of loss and accuracy. For the second question, we utilized a Local Interpretable Model-agnostic Explanations (LIME) method, to explain the predicted label, single scan at a time.

We plan to run a series of experiments: first running models on unaugmented data, then running chosen two models of augmented data and then running models by making variations to the data like changing contrast, varying gaussian noise and undersampling data using different strides. We use LIME to explain the label

prediction for each model and make inferences based on the results of LIME.

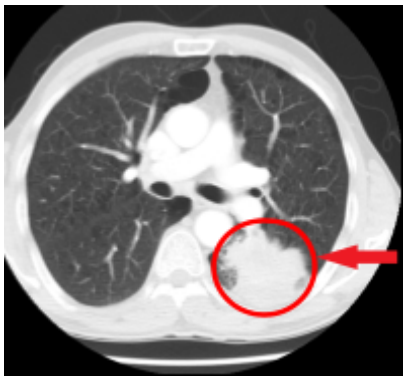

Figure 1. Adenocarcinoma on an axial CT scan.

II. Data

Our data set is the Chest CT-Scan Images Dataset uploaded to Kaggle by Mohamed Hany [6]. The dataset includes CT slices of non-cancerous chest tissue and three types of chest cancers: adenocarcinoma, large cell carcinoma, and squamous cell carcinoma. These files are not in the DICOM format or other typical imaging data formats but in Portable Graphic Network (.png) and Joint Photographic Experts Group (.jpeg) formats. This is beneficial for our use, as the files are not as massive as medical imaging files but retain the pixel intensities to tell the anatomical structures apart. The dataset was originally compiled to train convolutional neural networks to parse between healthy patients and patients with a form of chest cancer, but we are using this dataset because the listed cancer types are also forms of pulmonary cancer. Like with chest cancers, an abdominal CT is taken to determine if a patient has lung cancer. Thus, using this dataset will be applicable to our experimental goal of trying to parse between healthy patients and patients with pulmonary cancer, as well as identify tumor location. Currently, there are 38 notebooks on Kaggle that have used this dataset and the dataset itself has received 100% in all of the Kaggle Usability categories - Completeness, Credibility, and Compatibility.

Our team implemented the Kaggle file downloader in Google Colab to retrieve the CT dataset. The dataset was already split into testing (70%), training, (20%) and validation (10%) sets by the original data uploader when downloaded. We decided to keep these split percentages for the testing of our models. Using the os, glob, and fnmatch libraries, we counted the number of scans for each lung CT scan type - normal (no cancer), adenocarcinoma, large cell carcinoma, and squamous cell carcinoma - within each testing, training, and validation folder. An overview of the chest CT scan data can be found in Table 1.

Table 1. Summary of Chest CT-Scan Images Dataset

	Training	Testing	Validation	Total
Normal	148	54	13	215
Adenocarcinoma	195	120	23	338
Large Cell Carcinoma	115	51	21	187
Squamous Cell Carcinoma	155	90	15	260
Total	613	315	72	1000

Of the cancerous CT Scans, Adenocarcinmoca, Large Cell carcinoma, and Squamous Cell carcinoma are all classified as Non - Small Cell Lung Cancer Types. Non - Small Cell Lung Cancer Types (NCSLCs) are the most common type of lung cancer that originates from the cellular level [7]. They are cancers of the cell linings and lung surfaces, yet each type has unique features that distinguish them on CT.

Adenocarcinoma is the most common lung cancer type in the USA and is strongly associated with previous smoking. It's characterized by chronic inflammation and scarring and is usually located at the periphery of the lungs [8].

Large Cell Carcinoma (LCLC) is usually distinguished by its large peripheral mass and irregular margins, mostly identified on the outer lung edges [9].

Squamous Cell Carcinoma mostly develops along the airways of the lungs near the left and right

bronchus, and thus is primarily found centrally within the CT Scans.

The scans within the dataset are not taken from the same z-position (Figure 2), which will lend to training more robust models, as the model will not get trained on a single depth within the CT scan.

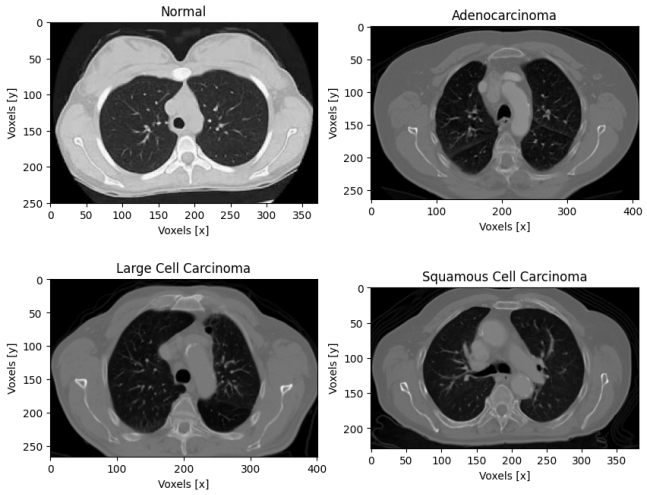

Figure 2. Sample scans from each label, healthy or with cancer, in the Chest CT scan dataset.

Rather than splitting the data amongst cancer types - as is the default splitting on Kaggle - we grouped all of the cancer CT scans together. This was done to help achieve our first aim, which is to develop a model to distinguish between cancerous and non-cancerous lung CT scans. Combining all of the cancerous CT scans created a larger training dataset for the models. Each scan was then labeled to be either "non-cancerous" or "cancerous."

Limitations of Data

The major limitation for this analysis is that data is limited. There are high chances of neural network based models memorizing the data. Explained later in this report, we achieved high accuracy for our complex neural networks with multiple layers. We address this limitation through data augmentation and conducting performance gradient experiments by changing contrast of images, introducing gaussian noise, and undersampling images using different strides.

Another limitation is that we don't have patient details or the coordinates of the tumor. Thus, we can only gauge the location of the tumor from the type of cancer it is. It was difficult to choose an interpretable technique to explain tumor location since we didn't know what features are important just on the basis of suggested location of the tumor through type of cancer. Due to this limitation, we chose a simple explainability technique called LIME which gave informative but varying results. LIME. Using LIME can help us determine what features of the image the models are using to make the classification and whether the features indeed align with the spatial location of cancer or whether the model is using erroneous features in the scan to classify the image. It does have a few limitations as discussed by Molnar (2020) in his book on Interpretable Machine Learning [10]. But for the scope of this project and considering data limitations, we decided to move forward with it.

III. Approach

For this project, we are assessing how well four algorithms can distinguish between healthy healthy and unhealthy patients with a form of pulmonary cancer. The algorithms we'll use for this task are *VGG16, 2D CNN, ResNet50, and DenseNet201*. We have decided to use these algorithms due to their popularity in image classification [11]. For each algorithm, the batch size was 32, the number of epochs was 10, and we used a 70/20/10 train/test/validation split.

A. Baseline Model

For our baseline model, we chose 2D CNN with seven layers: two Conv2D layers, two MaxPooling2D layers, one Dropout layer, and two Dense layers (Table 2). We used ReLU activation function and sigmoid function for the output of our model. We took the reference from a Kaggle notebook [12].

Table 2. 2D CNN Architecture

Layer	Output Shape	Param #
Input	(None, 460, 460, 3)	0
2D Convolution	(None, 460, 460, 8)	104
2D Max Pooling	(None, 230, 230, 8)	0
2D Convolution	(None, 230, 230, 16)	528
2D Max Pooling	(None, 115, 115, 16)	0
Dropout	(None, 115, 115, 16)	0
Flatten	(None, 211600)	0
Dense	(None, 300)	63480300
Dropout	(None, 300)	0
Dense (Output)	(None, 1)	301

We trained the model on the original dataset with two labels - cancerous and non-cancerous. The results of the analysis are in Table 6 in the Results Section.

We then decided to use the LIME explainability technique to see the results for a squamous cancerous image, which has cancer near the center of the lung using a trained 2D CNN model.

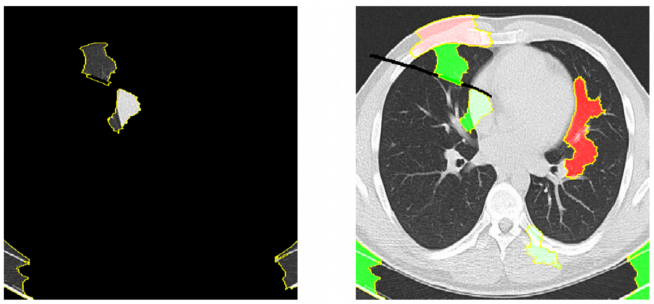

Figure 3. LIME result for 2D CNN model.

LIME Interpretation. In LIME, the color green is used to indicate features that positively correlate with the predicted outcome, while the color red is used to indicate features that negatively correlate with the predicted outcome. Since the cancer is in the center of the lung, for this and other models, we will interpret maximum green at the expected location of the cancer as a measure of good performance of the model in detecting the location of the cancer, which helps in addressing our second objective i.e., identifying the location of the cancer.

For 2D CNN, we observed that green is not near the center of the lung and we interpreted it as not being a good model as far as identifying cancer location is concerned. We understand that 85% accuracy is good, but that can be because of limitations of data which we discussed earlier.

B. VGG16 Model

Next, we created a neural net using a pre-trained VGG16 model with weights and biases borrowed from ImageNet. We referenced an existing Kaggle notebook that analyzed the same chest CT dataset [13]. Our resulting model had the first layer of VGG16 (fully connected layers of the VGG16 model not included), a flatten layer used to convert the output of the convolutional layers from a 3D tensor to a 1D tensor that can be passed to the fully connected layers (Table 3). Two other layers are a batch normalization layer and a dense output layer with a sigmoid activation function. The VGG16 layers are frozen to prevent their weights from being updated during training. Only the weights of the final layers (i.e., the Flatten, BatchNormalization, and Dense layers) will be updated during training.

Table 3. VGG16 Neural Net Architecture

Layer	Output Shape	Param #
Input	(None, 460, 460, 3)	0
VGG16 (functional)	(None, 512)	14714688
Flatten	(None, 512)	0
Batch Normalization	(None, 512)	2048
Dense (output)	(None, 2)	1026

The performance for this model can be seen in Table 6 in the Results section.

We then decided to use the LIME explainability technique to see the results for a squamous cancerous image, which has cancer near the center of the lung using a trained VGG16 model.

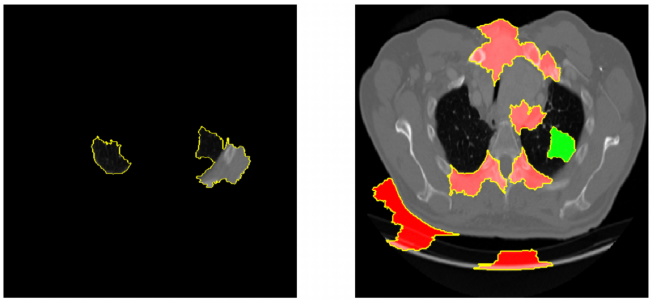

Figure 4. LIME result for squamous cancerous image for VGG16 model.

As presented in Figure 4, there is a substantial amount of red near the center of the squamous cancer, which is opposite to what we expected.

However, we wanted to observe the performance of this model on augmented data and see how the results vary when we use LIME for the VGG16 model trained on augmented data. Therefore, we selected VGG16 for training on augmented data.

C. ResNet50 Model

The third model is a neural network created with a pre-trained ResNet50 model. We took the reference from the Kaggle notebook [12]. The layers in the model are: ResNet50 model (fully connected not included), a dropout layer that randomly drops out some neurons during training to prevent overfitting, a flatten layer is used to convert the output of the convolutional layers from a 3D tensor to a 1D tensor, a batch normalization layer, another dropout layer and a dense layer with sigmoid activation function (Table 4). The layers for ResNet are frozen to prevent their weights from being updated during training. Only the weights in the last layers are updated during training.

Table 4. ResNet50 Neural Net Architecture

Layer	Output Shape	Param #
Input	(None, 460, 460, 3)	0
Resnet50 (functional)	(None, 2048)	23587712
Dropout	(None, 2048)	0
Flatten	(None, 2048)	0
Batch Normalization	(None, 2048)	8192
Dropout	(None, 2048)	0

The performance for this model can be seen in Table 6 in the Results section.

We then ran LIME for a squamous cancer image as before and observed the results

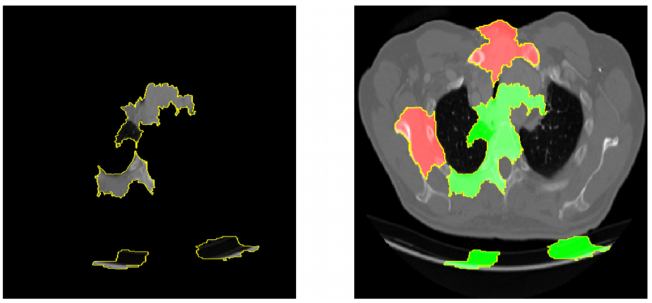

Figure 5. LIME result for squamous cancerous image for ResNet50 model.

As we can see, the green is mostly towards the center of the lung, which is expected for a squamous cell carcinoma CT scan. We can infer that ResNet50 did a decent job as far as LIME interpretation is concerned. Hence, we decided to choose this model for further analysis.

D. DenseNet201 Model

The fourth model is a neural network created with a pre-trained DenseNet201 model. We took the reference from the Kaggle notebook [12]. The layers in the model are: a pre-trained DenseNet201 model as the first layer (fully connected not included), a flatten layer is used to convert the output of the convolutional layers from a 3D tensor to a 1D tensor, a batch normalization layer, another dropout layer and a dense layer with softmax activation function (Table 5). The layers for DenseNet are frozen to prevent their weights from being updated during training. Only the weights in the last layers are updated during training.

Table 5. DenseNet201 Neural Net Architecture.

Layer	Output Shape	Param #
Input	(None, 460, 460, 3)	0
DenseNet201 (functional)	(None, 1920)	18321984
Flatten	(None, 1920)	0
Batch Normalization	(None, 1920)	7680
Dense (output)	(None, 2)	3842

We then ran LIME for a squamous cancer image as before and observed the results.

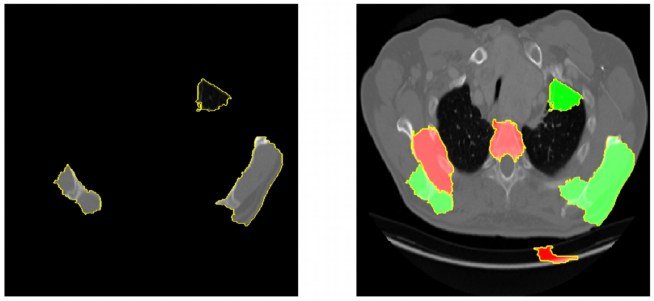

Figure 6. LIME result for squamous cancerous image for DenseNet201 model.

As we can see the LIME interpretability for DenseNet201, is unreliable despite the high accuracy, therefore we didn't move forward with further analysis on this model.

IV. Data Augmentation

To address the limitation of the data, we performed data augmentation. Based on previous analysis, ResNet50, VGG16 and DenseNet201 had almost similar performance on test data, with DenseNet201 being the best considering loss and accuracy. However, given that DenseNet has more than double the layers as compared to other models, chances of overfitting are more. Also, it didn't perform well with LIME. Therefore we decided to use VGG16 and ResNet for further analysis.

This data was augmented by cropping the images by 25 pixels, flipping 50% of all training images vertically, and applying a Gaussian blur to the images. Doing this doubled our training dataset from 613 images to 1226 images, validation dataset from 72 to 144, and test data from 315 to 630. An example of this augmentation is displayed in Figure 7.

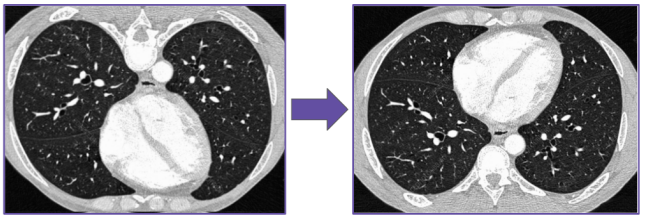

Figure 7. Example augmentation of data. Original CT scan is on the left and the augmented scan, flipped vertically, is on the right.

A. VGG16 Performance on Augmented Data

The performance of VGG dropped significantly when trained on augmented data as can be seen in Table 6 in the Results section.

Based on the results shown in Table 6, VGG didn't perform well after being trained on the augmented data. However, we also wanted to observe the results a poor performing model would give on LIME technique.

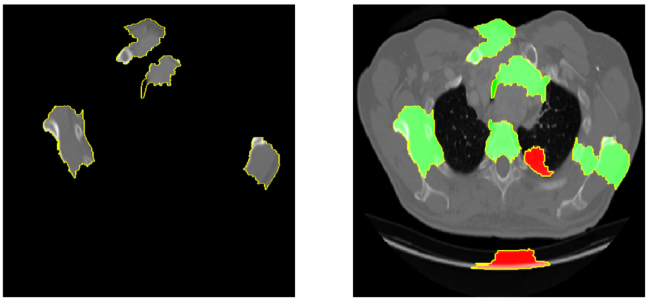

Figure 8. LIME result for squamous cancerous image for VGG16 model trained on augmented data

As seen in Figure 8, the green area is spread across the different parts of the lung including the center, where the location of cancer is for the squamous lung cancer type. We can only infer that VGG16 despite low accuracy as compared to before did better on LIME interpretability.

B. ResNet50 Performance on Augmented Data

ResNet50 performed very well on the augmented data. It gave steady results for train and validation data as can be seen in the figure below.

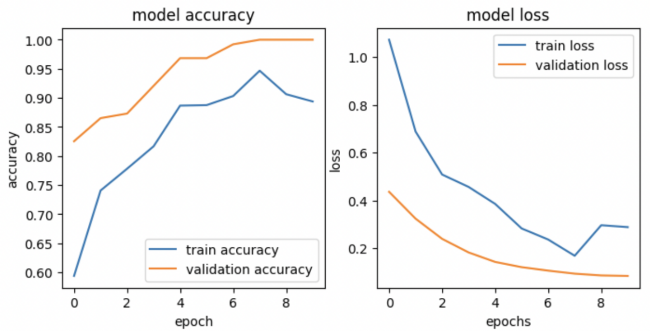

Figure 9. Accuracy and loss for ResNet50 on augmented train and validation data.

On test data, we got a pretty high accuracy of 0.9929 and loss of 0.1429. Overall, as far as the performance in label prediction is concerned, ResNet50 did a great job after being trained on augmented data and unaugmented data. Next, we decided to check the ResNet50 model trained on augmented data on LIME.

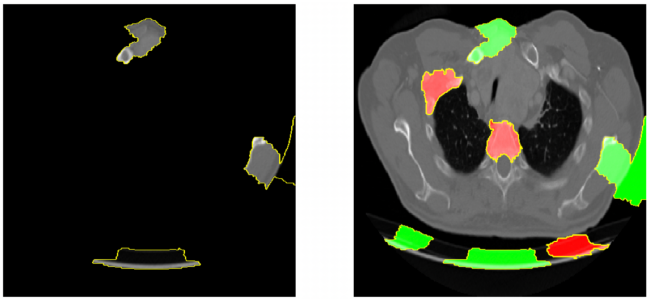

Figure 10. LIME result for squamous cancerous image for ResNet50 model trained on augmented data.

As displayed in the image above, the performance of LIME on the image was poor since the features that positively led to prediction as cancerous lung are not in the center of the image as expected for Squamous Lung Cancer.

C. Results

Table 6. Comparison of Model Evaluation Metrics

Model	Original, Unaugmented Data		Original and Augmented Data	
	Loss	Accuracy	Loss	Accuracy
VGG16	0.6152	0.9532	1.2348	0.2824
2D CNN	21.3331	0.8453	-	-
ResNet50	0.1494	0.9964	0.1429	0.9964
DenseNet	0.0616	0.9928	-	-

Based on the unexpected results that we got for both VGG16 and ResNet50 after being trained on the unaugmented data, we introspected the analysis and made following conclusions:

1. Since we didn't have patient information after we augmented the data, our dataset no longer adhered to the i.i.d (independent and identically distributed) assumption that we generally make for any machine learning problem.

2. LIME has many limitations in interpretability [14]: the explanations of similar examples can be different, LIME explanation fidelity can be low and explanation depends on the choice of hyperparameters.

The major reason behind these is the dataset we used for the analysis. Many classification algorithms utilize image segmentation during training, which often improves the accuracy of the algorithm used [15]. However, segmentations were not available for our dataset, nor did we have the time bandwidth to segment the images ourselves. Segmentation requires time and knowledge of the domain space (chest CT) to create an accurate representation of the lung space. Originally, our group wanted to use the LUNA16 dataset. However, upon trying to work with the data, we found that all of the scans were corrupted. Thus, we pivoted to the Chest CT dataset available on Kaggle, though it did not have any image segmentations to accompany the data. The inclusion of image segmentations in the training of our models in the future may improve model classification performance.

Also, using other techniques like saliency maps or SHapley Additive exPlanations (SHAP) can help us get more consistent interpretation as compared to LIME. For our future analysis, we can use these learnings to make better choices with dataset and interpretability techniques.

V. Performance Gradient

Overall, the models performed well on our first objective i.e., distinguishing between cancerous and non-cancerous lungs. However, since the accuracy was high, we decided to check the robustness of the models - VGG16 and ResNet50 by making changes to the contrast of images, introducing gaussian noise into the images and undersampling images using different strides.

A. Gaussian Noise

To evaluate how the addition of Gaussian Noise affects the model performance, we experimented with the application of Gaussian Noise variances between 0.5 to 4.00 inclusive on the test data images. Then we ran both ResNet50 and VGG16 for different gaussian noise variances to measure the difference in the performance.

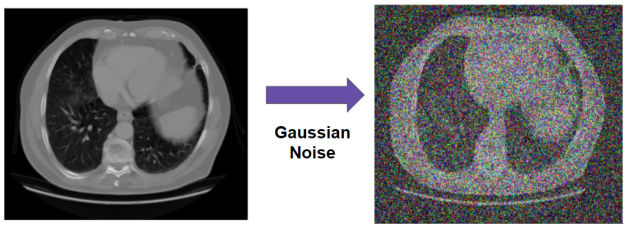

Figure 11. Application of gaussian noise on a CT scan.

As expected, the adding noise to the CT Scans decreased the model accuracy significantly for both models. However, unexpectedly, we observed that even with the most minimal addition of Gaussian Noise (variance = 0.5), the accuracy of the VGG16 model plummeted significantly to 0.20 and decreased towards 0.07 accuracy as the variance tended towards 4.00. While accuracy decreased significantly, the test loss remained relatively constant at approximately 0.83.

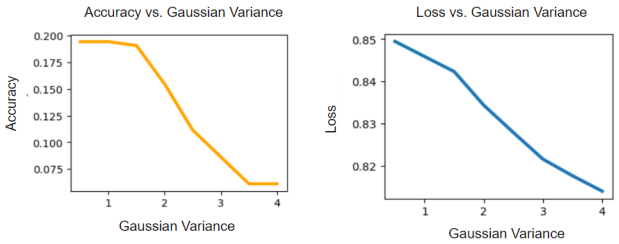

Figure 12. Accuracy and loss for VGG16 versus gaussian noise with variance range (0.5 - 4.00).

For the ResNet50 model, the accuracy dropped to 0.20 but unlike the VGG16 model, the accuracy remained relatively constant across the range of variances.

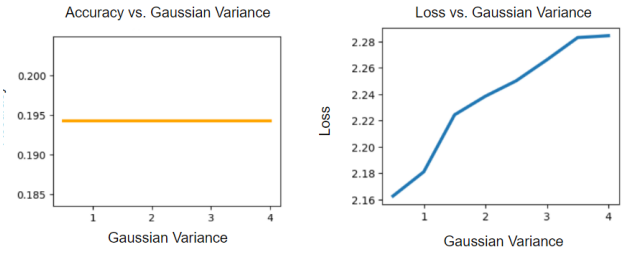

Figure 13. Accuracy and loss for ResNet50 versus gaussian noise variance range (0.5 - 4.00).

B. Contrast Variation

Since the contrast of the CT scan images can vary and is generally on the darker side, we chose the contrast range between 0 and 1. We modified test data images for different values of contrast variance (0.2, 0.4, 0.6, 0.8). Then we ran both ResNet50 and VGG16 for different contrast variations to measure the difference in the performance. The expectation was that as contrast reaches the value of 1, the accuracy for different models will reach closer to performance on the original dataset, since 1 is the contrast range for the original dataset.

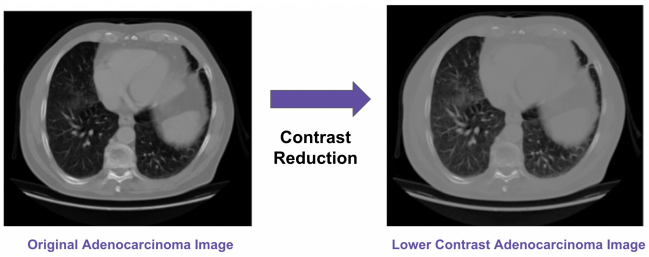

Figure 14. Contrast reduction on a CT scan.

For VGG16 the accuracy dropped significantly ~ 0.2 with the contrast change, as can be seen in the graphs below.

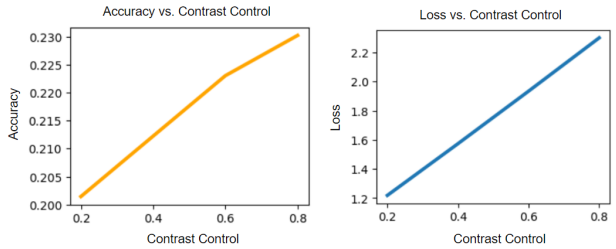

Figure 15. Accuracy and loss for VGG16 versus contrast range (0.2, 0.4, 0.6, 0.8).

For ResNet50, the accuracy varied between 0.88 to ~1.00 as contrast changed from 0.2 to 0.8.

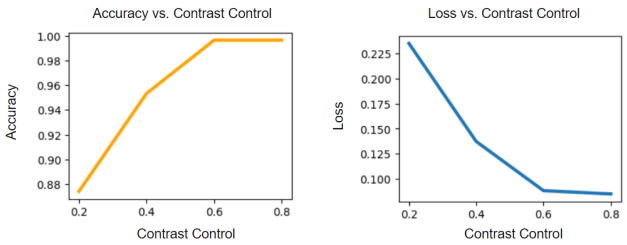

Figure 16. Accuracy and loss for ResNet50 versud contrast range (0.2, 0.4, 0.6, 0.8).

The overall observation is that VGG16 performance deteriorates on images as the contrast for the images change. The ResNet50 is evidently more stable as compared to the VGG16 on variation of contrast for CT Scan images. ResNet50 also met the expectation that we stated earlier for the experiment. There was also not a huge variation in the performance for ResNet50 with contrast changes and it can be inferred that it is robust for the given problem.

C. Undersampling

Undersampling with stride is a technique used in machine learning to reduce the size of a dataset by selecting a subset of data points with a fixed stride. For example, if we have a dataset with 1000 data points and we want to reduce it to 500 data points with a stride of 2, we would select every other data point starting from the first one. This means we would select data points 1, 3, 5, 7, and so on, until we reach the 500th data point.

We modified test data images for different values of strides (2, 4, 6, 8, 10). Then we ran both ResNet50 and VGG16 for different strides to measure the difference in the performance. The expectation was that as stride increases the accuracy should decrease, since increasing stride leads to loss of information within an information that generally makes prediction difficult. Also, the maximum value of stride is 10, which will lead to major impact of important features of an image.

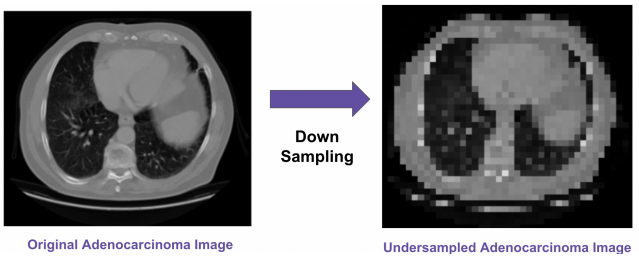

Figure 17. Undersampling on a CT scan.

VGG16 performed opposite to the expectation, as the stride increases the loss decreases and accuracy increases.

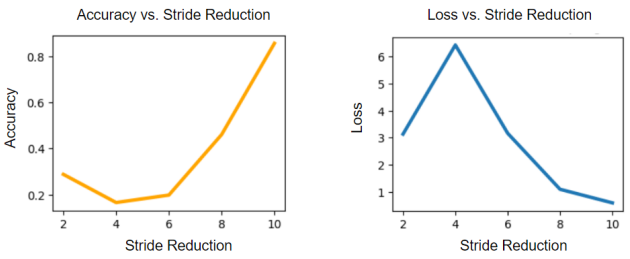

Figure 18. Accuracy and loss for VGG16 versus undersampling stride (2, 4, 6, 8, 10).

ResNet50 performed as expected, the accuracy decreased with the increase in stride and loss increased.

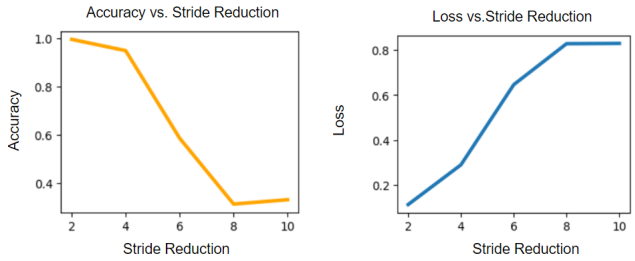

Figure 19. Accuracy and loss for ResNet50 versus undersampling stride (2, 4, 6, 8, 10).

The overall observation is that VGG16 performance becomes better on images as the stride for undersampling of the images increases. Further analysis might be required to make solid inference about this unexpected behavior of the model. The ResNet50 performed as expected since performance deteriorates with stride increase. There was also not a huge variation in the performance for ResNet50 with undersampling and it can be inferred that it is robust for the given problem.

D. Overall Results

From the gradient performance experiments, we found that generally, manipulating the images led to a reduction in the model's performance. Specifically, applying any degree of Gaussian Noise Variance to the images destroys our high accuracy although we observed that the VGG16 and ResNet50 models performed differently as the variance was introduced to the test images. However, via the Contrast Control experiment, ResNet50 predicted the image classes with 88% accuracy even when the image was dimmed to a contrast of 0.5. This is a positive result indicating that given CT Scans from different machines or contrast settings, the model still performs well. Then as expected, as the image contrast was brought back to the original images' form, the accuracy of the ResNet50 model returned to its original performance. This observation was greatly juxtaposed by the VGG16 model performance where the accuracy dropped significantly when the image contrast was reduced and failed to perform at its peak accuracy even when the image contrast was returned to its original form. Lastly, by performing an image resolution reduction through the removal of different row and column strides, we expected that as the image resolution decreases, both model accuracies will decrease. While this trend was correctly observed in the ResNet50 model, we found that reducing image resolution increased model accuracy for VGG16. From these observations, we conclude that most times, the ResNet50 model produced expected results, while the VGG16 model behaved in unexpected manners. This leads us to believe that the ResNet50 model is more robust than the VGG16 model for the given data. Further analysis might be required to make any strong conclusions.

VI. Our Results and Related Work

Our VGG16 model performed similarly to other VGG16 models in literature; in a study by Ramanjaneyulu et al., a dataset consisting of lung nodules - the LUNA16 dataset - was used to train a VGG16 for the detection of lung cancer, where the nodules, though benign in nature, can be used to detect malignant cell masses [15]. From their study, the research group found that on the LUNA16 dataset alone - that is, without augmentation - their model achieved a 99.84% training accuracy, which is comparable to our 95.32% training accuracy pre-data augmentation.

Our 2D CNN model performance is similar to that of another group; a study by Song, Zhao, Luo, and Dou sought to classify lung nodules on CT images through the use of CNN, deep neural network (DNN), and stacked autoencoder (SAE) algorithms [16]. Like in our study, this group only used one dataset - the Lung Image Database Consortium and Image Database Resource

Initiative (LIDC-IDRI) dataset - and thus had access to 9,106 images for training their models. While this is a large biomedical dataset, the size of the dataset is small by typical machine learning standards [17]. The group actually reduced the amount of data available to them by shrinking each image to include only the lung nodule, reducing the image sizes to 28 × 28 pixels. After training their model, the group found that their CNN performed the best of all their models with an accuracy of 84.15%, which is comparable to our model. The performances of our models with the original, unaugmented data and the model from Song, Zhao, Luo, and Dou can be improved in future work by including additional CT scan datasets, namely the LUNA16 and the LIDC-IDRI datasets [18,19]. Adding these datasets, and including additional augmented data to the training set, would improve the performance of our 2D CNN, ResNet50, and DenseNet201 models; this is because as more layers are added in a neural network, the number of trainable parameters increases, which increases the need for data to prevent underfitting of the model [17]. This is supported by work from Lin, Jeng, and Chen, who investigated the use of a 2D CNN model for the identification of cancer on CT images [20].

In their work, the research group used datasets from the International Society for Optics and Photonics with the support of the American Association of Physicists in Medicine (SPIE-AAPM) Lung CT Challenge and LIDC-IDRI datasets. Each dataset consists of "22,489 CT and 244,617 CT scans 512-by-512 pixels in size, respectively." Like in our exploratory analysis, their dataset included both healthy patients and patients with some form of pulmonary cancer. Prior to parameter tuning, Lin, Jeng, and Chen obtained an average accuracy rate of 91.97% on the LIDC-IDRI dataset and an accuracy rate of 94.68% on the SPIE-AAPM dataset. The architecture of the Lin, Jeng, and Chen model includes an input layer of a 50-by-50 pixel image, two convolution, ReLU, and max-pooling blocks with kernel sizes 5-by-5 for the convolutional layer and 2-by-2 for the max-pooling layer, two fully connected layers, a softmax layer, and finally a classification layer. Our 2D CNN instead had an input image size of 460-by-460 pixels, had a kernel size of 2-by-2, and included dropout, dense, and flattening layers (Table 2). In nearly all machine learning applications, kernel size is chosen to be odd to minimize distortions between layers from aliasing [21,22]. However, machine learning enthusiasts have stated that whether a kernel has even or odd dimensions does not impact algorithm performance with convolutional neural networks [23].

Compared to our 2D CNN, which obtained an accuracy of 84.53% (Table 2), the Lin, Jeng, and Chen model definitely outperformed us; this is in part due to the fact that they were able to suitably train the parameters in their model because they included more CT scans in their training sets. Because the performance of the Lin group 2D CNN differs from the Song group CNN, even on the same dataset, the structure of the architecture plays a role as well; the Lin group does not utilize a dropout layer like the Song group and the Lin group has larger images as inputs, but the architecture of both CNNs are identical beyond those two parameters. Our 2D CNN is more similar to the Lin group 2D CNN model, so there could be a performance increase with the introduction of more data.

Moving onto the performance of our ResNet50 model compared to the literature, our model obtained a training accuracy of 99.64%, while a pre-trained ResNet50 algorithm in a study conducted by Mohammed and Cinar obtained a 97.05% training accuracy [24]. Mohammed and Cinar used the SPIE-AAPM dataset and augmented data for training their models through rotation of the images; that is, they increased their dataset seven-fold by rotating the images by 45,

90, 135, 180, 225, 270, and 315 degrees. The reasoning for rotation for augmentation and not adding noise into the images was that the research group wanted to preserve image quality for training their models. Though Mohammed and Cinar had a vastly larger dataset than our group, it is difficult to compare the performances of our models directly, as we trained our models with noisy and contrast-adjusted images in addition to images of good quality. Other differences between our ResNet50 model and that used by Mohammed and Cinar is that our model trained for 10 epochs, while theirs trained for 20, and that we had a batch size of 32 while they had a batch size of 64. One reason why our model may have performed better than the ResNet50 model of Mohammed and Cinar is that as batch size increases, the likelihood of the model becoming stuck in a local minima, rather than finding the global minimum, increases [25]. Additionally, larger batch sizes make a model less generalizable. It is possible that the model trained by Mohammed and Cinar became stuck in a local minima and, therefore, did not perform as well as out model. Our model is also more robust than their model; though it was trained with less data, our model was trained with a more diverse set of CT scans, allowing the model to parse through "difficult" data to best classify which CT scans contain cancer and which CT scans belong to healthy patients.

In another study by Sajja, Devarapalli, and Kalluri, the accuracy of a ResNet50 model was again evaluated when parsing between healthy and cancerous CT scans [26]. The group used the LIDC dataset and did not add any augmented data to their training dataset. However, the group did pre-process the LIDC dataset by removing slices from the CT scans and converting them into .jpeg format, which is similar to our dataset from Kaggle containing .jpeg and .png CT scan slices. The research group modified the ResNet50 algorithm to be binary and thus classify a scan as containing or not containing cancer. Their ResNet50 model, trained on 80% of the data, obtained an accuracy of 97.42%, which is comparable to our ResNet50 model accuracy.

Our DenseNet201 model had the second-highest accuracy of all our models at 99.28% and with the lowest loss at 0.0616; in a paper by Aashka Mohite, using the Iraq-Oncology Teaching Hospital/National Center for Cancer Diseases (IQ-OTH/NCCD) lung cancer dataset with augmented data, they achieved a training accuracy of 93% for their DenseNet201 model [27]. While the accuracies between our models are similar, the Mohite DenseNet201 model was pre-trained, accepted smaller images of a 224-by-224 size, and their data augmentation involved only cropping, zooming, translating, and rotating the images, not introducing noise or downsampling the images. Though the base IQ-OTH/NCCD dataset was larger than our Kaggle set - ours with a base of 1000 images and theirs with a base of 1190 images - our model likely performed better as it was able to learn features with noise and because we accepted more information during our input with our 460-by-460 pixel images. Similarly, in another paper by Manop Phankokkruad, using the Lung and Colon Cancer Histopathological Image (LCCHI) dataset and a DenseNet201 algorithm, detected lung cancer with an 89% accuracy rate. The LCCHI dataset includes 750 pulmonary CT scans; our DenseNet model may have performed better due to having a larger base dataset of 1000 scans. Not only did other researchers use more data, but they also were able to leverage segmentation in the training of their models.

VII. Conclusion and Future Work

Overall, through this exploratory analysis, we have deduced that the ResNet50 algorithm is the most robust to image quality variation. However, if the image quality is not an issue, a VGG16 algorithm will classify parts of an image with the highest accuracy.

In summary, the limitations of this project include the time constraint, the lack of image segmentation, and an insufficient CT scan image repository size. Additionally, the lack of demographic information we had for the images limited our ability to conduct meaningful exploratory analysis. If we had data concerning patient sex, age, and country of residence, we would have been able to produce more complex hypotheses that incorporated the different variables. Additionally, our interpretability methods of the predictions are limited by our use of LIME exclusively. Implementing saliency maps or SHAP would help produce a more consistent interpretation as compared to LIME. Given more time and greater computational resources, we would have trained our models on a larger number of epochs, similar to how other scientists have approached this problem. Additionally, we would have sought out a greater number of CT Scan images to increase our data set.

ACKNOWLEDGEMENTS

Student	Contribution
Reem Fashho	100
Amanda Nowacki	100
Shreya Shukla	100

REFERENCES

[1] *Causes of death - Our World in Data*. (n.d.). Retrieved April 21, 2023, from https://ourworldindata.org/causes-of-death

[2] *Lung cancer statistics | World Cancer Research Fund International*. (n.d.). Retrieved April 21, 2023, from https://www.wcrf.org/cancer-trends/lung-cancer-statistics/

[3] Mackintosh, J. A., Marshall, H. M., Yang, I. A., Bowman, R. v., & Fong, K. M. (2014). A retrospective study of volume doubling time in surgically resected non-small cell lung cancer. *Respirology*, *19*(5), 755–762. https://doi.org/10.1111/resp.12311

[4] del Ciello, A., Franchi, P., Contegiacomo, A., Cicchetti, G., Bonomo, L., & Larici, A. R. (2017). Missed lung cancer: When, where, and why? In *Diagnostic and Interventional Radiology* (Vol. 23, Issue 2, pp. 118–126). AVES Ibrahim Kara. https://doi.org/10.5152/dir.2016.16187

[5] Doi, K. (2007). Computer-aided diagnosis in medical imaging: Historical review, current status and future potential. *Computerized Medical Imaging and Graphics*, *31*(4–5), 198–211. https://doi.org/10.1016/j.compmedimag.2007.02.002

[6] *Chest CT-Scan images Dataset | Kaggle*. (n.d.). Retrieved April 21, 2023, from https://www.kaggle.com/datasets/mohamedhanyyy/chest-ctscan-images

[7] *Non-Small Cell Lung Cancer > Fact Sheets > Yale Medicine*. (n.d.). Retrieved April 25, 2023, from https://www.yalemedicine.org/conditions/non-small-cell-lung-cancer#:~:text=What%20is%20non%2Dsmall%20cell,surface%20of%20the%20lung%20airways

[8] Zhang, J., & Yang, G. C. H. (2012). Adenocarcinoma. In *Lung and Mediastinum Cytohistology* (pp. 122–144). Cambridge University Press. https://doi.org/10.1017/CBO9781139023351.007

[9] Panunzio, A., & Sartori, P. (2020). Lung Cancer and Radiological Imaging. *Current Radiopharmaceuticals*, *13*(3), 238–242. https://doi.org/10.2174/1874471013666200523161849

[10] Interpretable Machine Learning - Christoph Molnar. (n.d.). Retrieved April 25, 2023, from https://christophmolnar.com/books/interpretable-machine-learning/

[11] *Image Classification Techniques. Image classification refers to a… | by Kavish Sanghvi | Analytics Vidhya | Medium*. (n.d.). Retrieved April 21, 2023, from https://medium.com/analytics-vidhya/image-classification-techniques-83fd87011cac

[12] *Chest Cancer Classification 90%+ on Test Set | Kaggle*. Retrieved April 21, 2023,(n.d.).https://www.kaggle.com/code/magedmahmoud/chest-cancer-classification-90-on-test-set.

[13] *Chest Cancer Classification With VVG 16 | Kaggle*. (n.d.). Retrieved April 21, 2023, from https://www.kaggle.com/code/yasserhessein/chest-cancer-classification-with-vvg-16

[14] *What's Wrong with LIME. While being one of the most popular… | by Denis Vorotyntsev | Towards Data Science*. (n.d.). Retrieved April 25, 2023,from https://towardsdatascience.com/whats-wrong-with-lime-86b335f34612

[15] Ramanjaneyulu, K., Kumar, K. H., Snehith, K., Jyothirmai, G., & Venkata Krishna, K. (2022). Detection and Classification of Lung Cancer Using VGG-16. *Proceedings of the 2022 International Conference on Electronic Systems and Intelligent Computing, ICESIC 2022*, 69–72. https://doi.org/10.1109/ICESIC53714.2022.9783556

[16] Song, Q. Z., Zhao, L., Luo, X. K., & Dou, X. C. (2017). Using Deep Learning for Classification of Lung Nodules on Computed Tomography Images. *Journal of Healthcare Engineering*, *2017*. https://doi.org/10.1155/2017/8314740

[17] *How Much Data Is Required for Machine Learning? – PostIndustria*. (n.d.). Retrieved April 21, 2023, from https://postindustria.com/how-much-data-is-required-for-machine-learning/

[18] *Data - Grand Challenge*. (n.d.). Retrieved April 21, 2023, from https://luna16.grand-challenge.org/Data/

[19] *Data From LIDC-IDRI*. (n.d.). https://doi.org/10.7937/K9/TCIA.2015.LO9QL9SX

[20] Lin, C.-J., Jeng, S.-Y., & Chen, M.-K. (2020). Using 2D CNN with Taguchi Parametric Optimization for Lung Cancer Recognition from CT Images. *Applied Sciences*, *10*(7), 2591. https://doi.org/10.3390/app10072591

[21] *Significance of Kernel size. Why the kernel size should be odd? What… | by Anuja Ihare | Analytics Vidhya | Medium*. (n.d.). Retrieved April 25, 2023, from https://medium.com/analytics-vidhya/significance-of-kernel-size-200d769aecb1

[22] *Why is Odd sized kernel preferred over Even sized kernel? | by Prasant Kumar | Geek Culture | Medium*. (n.d.). Retrieved April 25, 2023, from https://medium.com/geekculture/why-is-odd-sized-kernel-preferred-over-even-sized-kernel-a767e47b1d77

[23] *When do we use an even size kernel in convolutional neural network and why? - Cross Validated*. (n.d.). Retrieved April 25, 2023, from https://stats.stackexchange.com/questions/366739/when-do-we-use-an-even-size-kernel-in-convolutional-neural-network-and-why

[24] *View of Lung cancer classification with Convolutional Neural Network Architectures*. (n.d.). Retrieved April 25, 2023, from

https://journal.qubahan.com/index.php/qaj/article/view/33/20

[25] *How does Batch Size impact your model learning | by Devansh- Machine Learning Made Simple | Geek Culture | Medium*. (n.d.). Retrieved April 25, 2023, from https://medium.com/geekculture/how-does-batch-size-impact-your-model-learning-2dd34d9fb1fa

[26] Sajja T.K., Davarapalli R.M., and Kalluri H.K. (2019). *Lung Cancer Detection Based on CT Scan Images Using Deep Transfer Learning.* International Information and Engineering Technology Association (Vol. 36, Issue 4, pp. 339-344). https://doi.org/10.18280/ts.360406

[27] Mohite A. (2021). "Application of Transfer Learning Technique for Detection and Classification of Lung Cancer using CT Images." International Journal of Scientific Research and Management (Vol. 09, Issue 11, pp. 621-632). https://doi.org/10.18535/ijsrm/v9i11.ec02