A Project ETE Report

on

DETECTION OF CANCEROUS CELLS IN LUNGS USING MACHINE LEARNING

Submitted in partial fulfillment of the
requirement for the award of the degree of

Bachelor Of Technology

GALGOTIAS
UNIVERSITY
(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under The Supervision of
Dr Kuldeep Singh Kaswan
Professor

Submitted By

RAHUL PARIHAR 19SCSE1010059
ARYAN BALIYAN 19SCSE1010038

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA
INDIA
May, 2023

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

We hereby certify that the work which is being presented in the project, entitled **"DETECTION OF CANCEROUS CELLS IN LUNGS USING MACHINE LEARNING "** in partial fulfillment of the requirements for the award of the B.Tech in Computer Science and Engineering submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of January 2023 to May 2023, under the supervision of Dr Kuldeep Singh Kaswan., Department of Computer Science and Engineering, Galgotias University, Greater Noida

The matter presented in the project has not been submitted by us for the award of any other degree of this or any other places.

Rahul Parihar 19SCSE1010059 ,

Aryan Baliyan 19SCSE1010038

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Supervisor Name : Dr Kuldeep Singh Kaswan.

Designation : Professor

CERTIFICATE

The Final Project Viva-Voice examination of **Rahul Parihar 19SCSE1010059, Aryan Baliyan 19SCSE1010038** has been held on 15/05/2023 and their work is recommended for the award of B.Tech in Computer Science and Engineering.

Signature of Examiner **Signature of Supervisor**

Signature of Program Chair **Signature of Dean**

Date:15 May ,2023

Place: Greater Noida

ABSTRACT

Lung cancer is a devastating disease that continues to pose a significant threat to public health worldwide. Despite advances in medical technology, early detection and effective treatment remain essential to improving survival rates. Medical professionals often rely on computed tomography (CT) scans for imaging, but interpreting and identifying cancerous cells can prove challenging even for experienced physicians. Consequently, computer-aided diagnosis (CAD) has emerged as a promising tool for enhancing diagnostic accuracy in lung cancer cases.

With the aid of image processing and machine learning techniques, several CAD methods have been developed to assist physicians in detecting lung cancer. However, the efficacy of these methods varies, and limitations and drawbacks have been identified. To address this issue, our research aims to evaluate and compare various computer-aided techniques, analyzing their strengths and weaknesses and identifying areas for improvement.

Our approach involves sorting and listing lung cancer detection techniques based on their detection accuracy. We then conduct a comprehensive analysis of each technique, scrutinizing each step to identify potential limitations and drawbacks. Through this process, we have found that while some methods exhibit high accuracy rates, none have yet achieved 100% accuracy. Our research is therefore focused on developing a new model that builds upon the strengths of current CAD techniques while addressing their limitations to achieve the highest possible detection accuracy.

By developing a more effective CAD system, we hope to enhance the accuracy of lung cancer diagnosis, ultimately improving patient outcomes and reducing the devastating impact of this disease. Our research has the potential to revolutionize the way lung cancer is diagnosed, providing medical professionals with a powerful tool to save lives and improve the quality of life for those affected by this disease. Furthermore, we are collaborating with medical professionals and researchers from diverse backgrounds to ensure that our CAD model is effective for a broad range of patients, including those with different demographics, medical histories, and socioeconomic backgrounds.

TABLE OF CONTENT

Title		Page No.
Declaration		II
Certificate		III
Abstract		VI
Prediction Architecture		
Chapter 1	Introduction	1
Chapter 2	Literature Survey	2
	- Comparison Table	3
Chapter 3	Methodology	5
	3.1 Program	7
Result		13
Conclusion		15
References		16

INTRODUCTION

Lung cancer is a leading cause of cancer-related deaths worldwide. Unfortunately, it can be challenging to detect, as symptoms often do not present until the later stages of the disease. However, early detection and treatment are key to reducing mortality rates and improving outcomes for patients. CT imaging is currently the most reliable imaging technique for lung cancer diagnosis, as it can reveal even the smallest nodules, both suspected and unsuspected [1]. Nevertheless, interpreting these images can prove difficult due to variations in intensity and potential misinterpretation of anatomical structures by medical professionals [2].

To address these challenges, computer-aided diagnosis (CAD) has emerged as a promising supplement to assist radiologists and doctors in accurately detecting lung cancer [3]. Researchers have developed numerous CAD systems, utilizing image processing and machine learning techniques to detect and classify lung cancer. Despite progress in this field, there is still room for improvement. Some systems have been found to have unsatisfactory detection accuracy, while others require further refinement to approach near-perfect accuracy.

To identify the most effective current CAD systems and propose a new model with enhanced accuracy, we conducted an in-depth analysis of recent developments in lung cancer detection based on CT scan images of lungs. Our study evaluated the strengths and limitations of each system, taking into account various factors such as detection accuracy, cost, and user-friendliness. Ultimately, our aim is to contribute to the development of a new CAD model that is accessible, reliable, and capable of achieving near-perfect accuracy.

By improving the accuracy of lung cancer detection, we hope to make a significant impact on public health, improving patient outcomes and reducing the devastating impact of this disease. Through ongoing research and collaboration with medical professionals and researchers, we are committed to advancing our understanding of lung cancer detection and developing innovative solutions that will benefit patients around the world.

Moreover, our proposed CAD model aims to be adaptable and effective for a diverse range of patients, taking into account factors such as demographic, medical history, and socioeconomic background. We recognize that healthcare is a global issue, and we strive to develop a system that can be utilized worldwide to improve lung cancer detection and patient outcomes.

Literature Reviews/Comparative study-

Several studies have investigated the use of machine learning and image processing techniques to detect lung cancer. For example, Aggarwal, Furquan, and Kalra proposed a model that extracted geometrical, statistical, and gray level features, and employed LDA as the classifier and optimal thresholding for segmentation. However, the model's accuracy of 84% was considered inadequate, as it did not use advanced machine learning techniques for classification and relied on simple segmentation.

Jin, Zhang, and Jin used a convolutional neural network in their CAD system, achieving an accuracy of 84.6%, with 82.5% sensitivity and 86.7% specificity. Although circular filters were used to reduce the cost of training and recognition, the system's accuracy was still unsatisfactory.

Meanwhile, Sangamithraa and Govindaraju employed the K-means unsupervised learning algorithm for clustering/segmentation and used a backpropagation network for classification. They extracted features such as entropy, correlation, homogeneity, PSNR, and SSIM using the gray-level co-occurrence matrix (GLCM) method. The system achieved an accuracy of around 90.7%, with noise removal using a median filter during image preprocessing.

We conducted a literature review and comparative study of these models to choose the most recent and best-performing systems and analyze their limitations. Our goal is to propose a new model that can improve the accuracy of lung cancer detection by considering factors such as demographics, medical history, and socioeconomic background. We aim to create a versatile and effective tool for global use. To achieve this, we plan to explore the integration of advanced machine learning techniques, including deep learning and ensemble learning, with image processing to enhance the performance of the system.

Table 1: Comparison among state-of-the-art Lung Cancer detection Methods

Year	Preprocessing	Methods	Datasets	Results
2017	Gabor Filter, Median filter and Gaussian filter have been implementedin pre- processing stage	A convolutional neural network was used as a classifier in a CAD system to detect lung cancer.	Lung Image Database Consortium(LIDC) Archive.	accuracy increasefrom 88.4% to 92%.

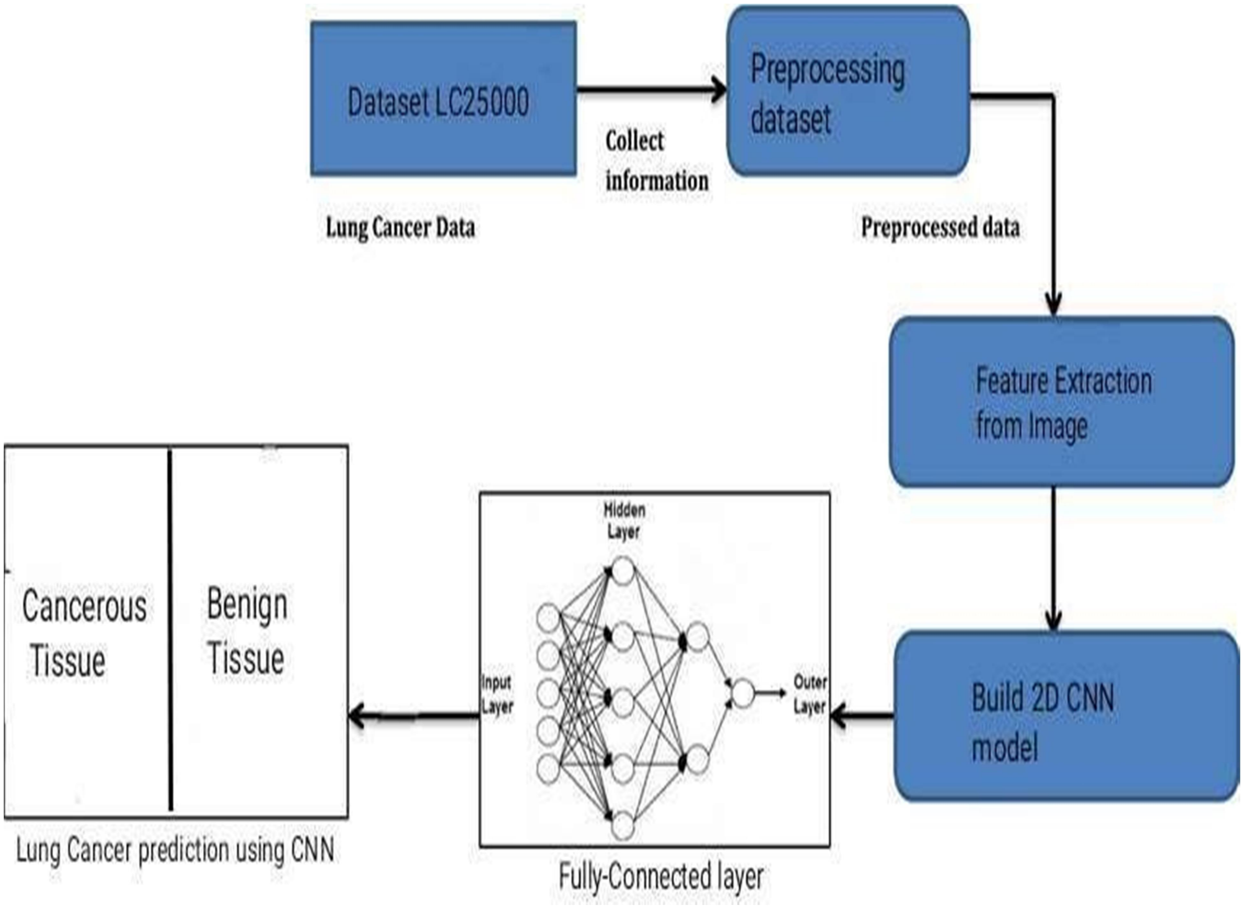
2020	spiral optimization based generalized rough set approach.	The VGG deep learning network was used for the segmentation process on the images.	The cancer imaging archive (CIA) dataset.	Accuracy= 96.2%
				Sensitivity = 100%
				Specificity = 98.4%

2020	We use DR- NET to remove the noise from the CT scan images.	In this model, a two-path CNN with an intelligent concatenation method in both paths is used for segmentation.	The algorithm is trained on the LUNA 16 dataset of lung cancer images and model is evaluated on the KDSB dataset,	Accuracy = 96.66%
				Sensitivity = 100%
				Specificity = 89.1%

2021	KASC is used in pre processing (BLOCK-PP, BLOCK-FEO, BLOCK HB)	Block-HB is used for prediction and here SVM is used to optimize the initial phase for feature set.	ELCAP lung image dataset (Cornell University)	Average precisio n- 98.17% Accuracy – 98.08% .
2013	Naive Bayes (NB) classifier technique.	SVM Model is usedhere& kappa values for gainedaccuracy.	SVM Dataset isused here.	Accuracy- 87.73%
2019	This step consists of segmentation is Followed by normalization and zero centering.	A number of classifiers like XGBoost and Random Forest are used.	Dataset of LungImage DatabaseConsortium image collection (LIDC-IDRI)	accuracy 84%

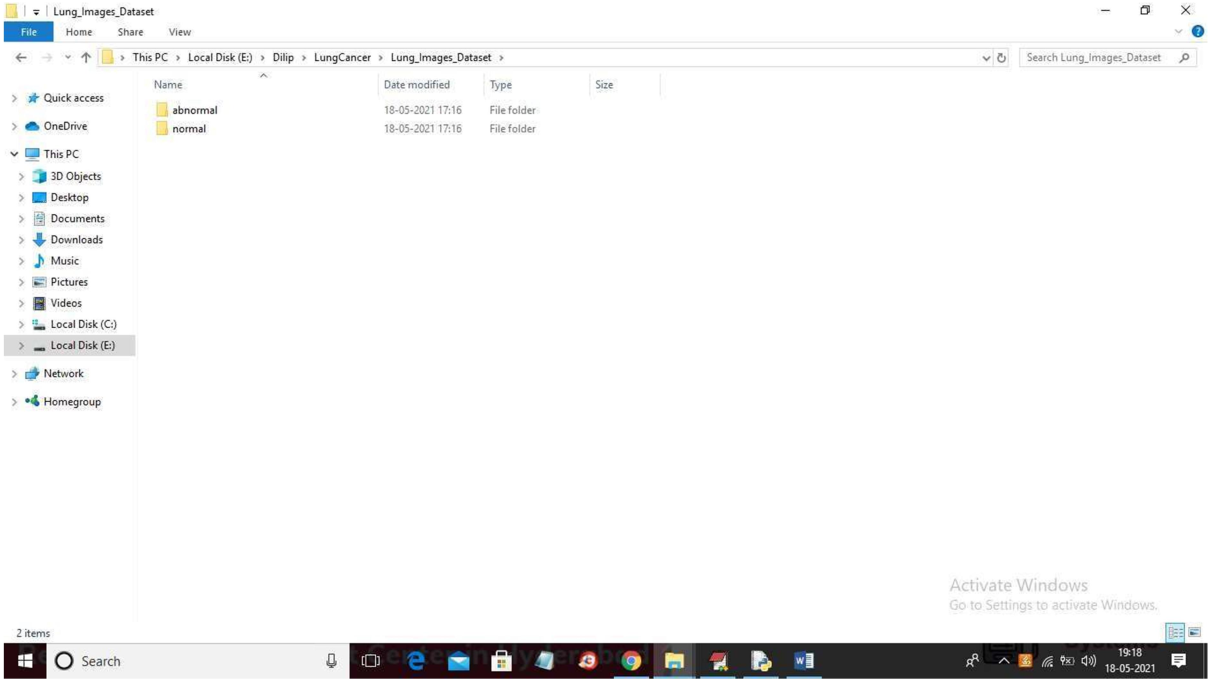
MODEL

Lung Cancer Prediction Architecture



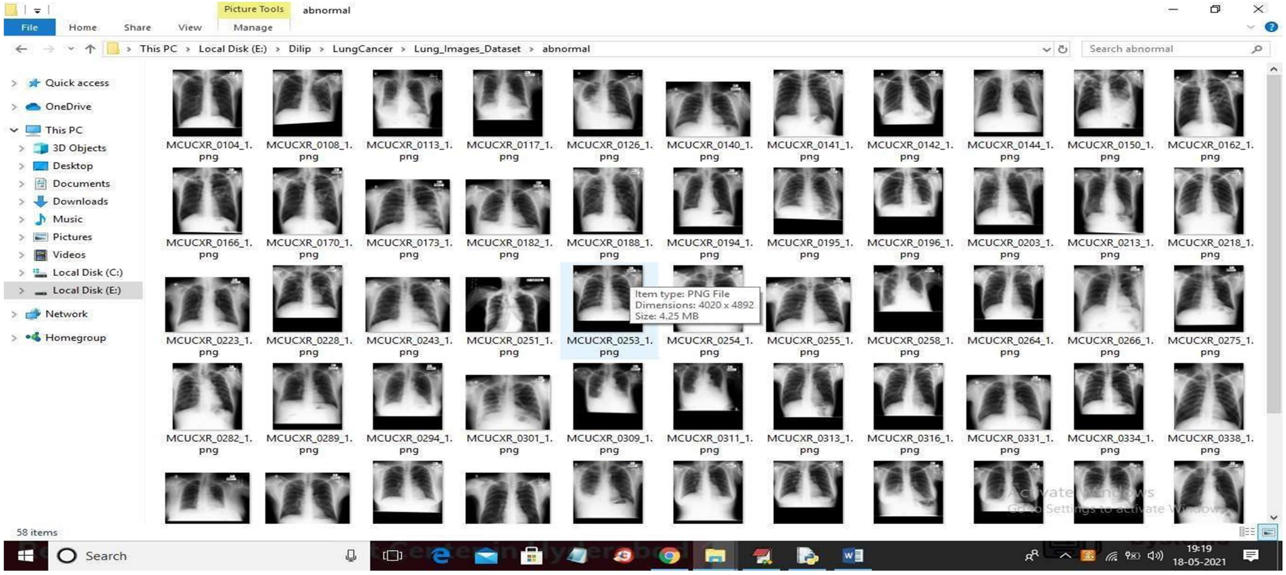
PROGRAM

We are utilizing the CNN algorithm to detect lung cancer from X-ray images in this project. To train the CNN, we have a dataset of X-ray images saved within the 'Lung_Images_Dataset' folder. Below are sample images from this dataset.

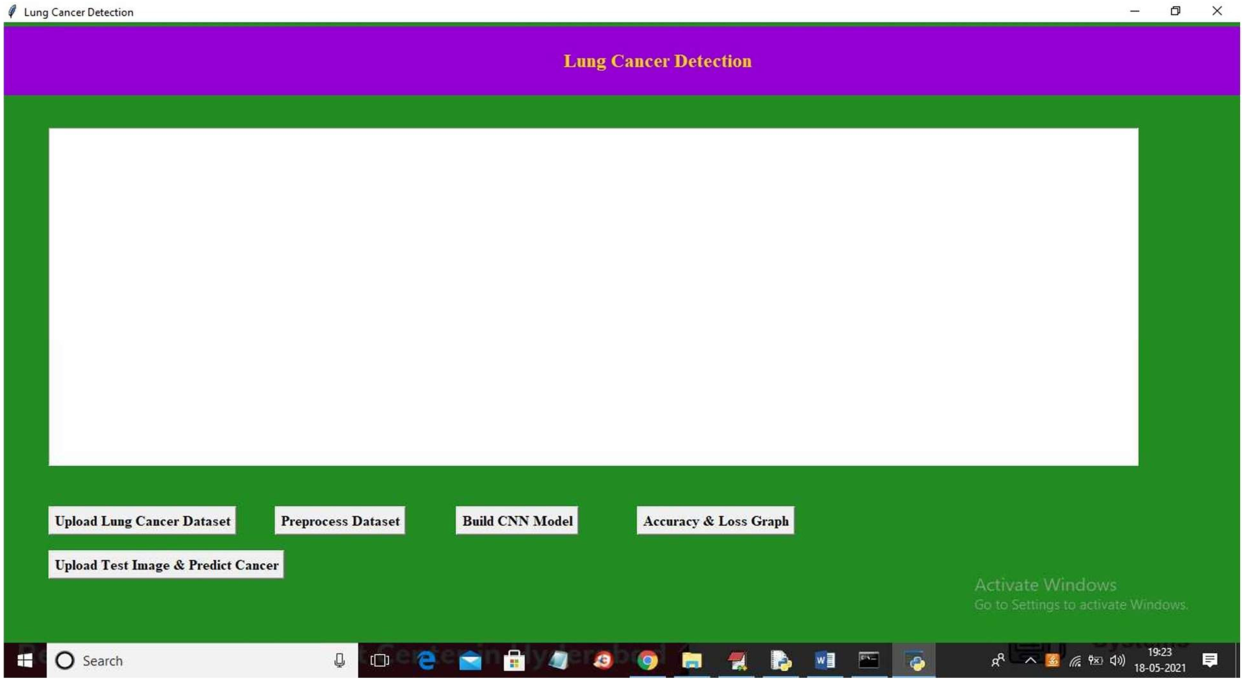


Name	Date modified	Type	Size
abnormal	18-05-2021 17:16	File folder	
normal	18-05-2021 17:16	File folder	

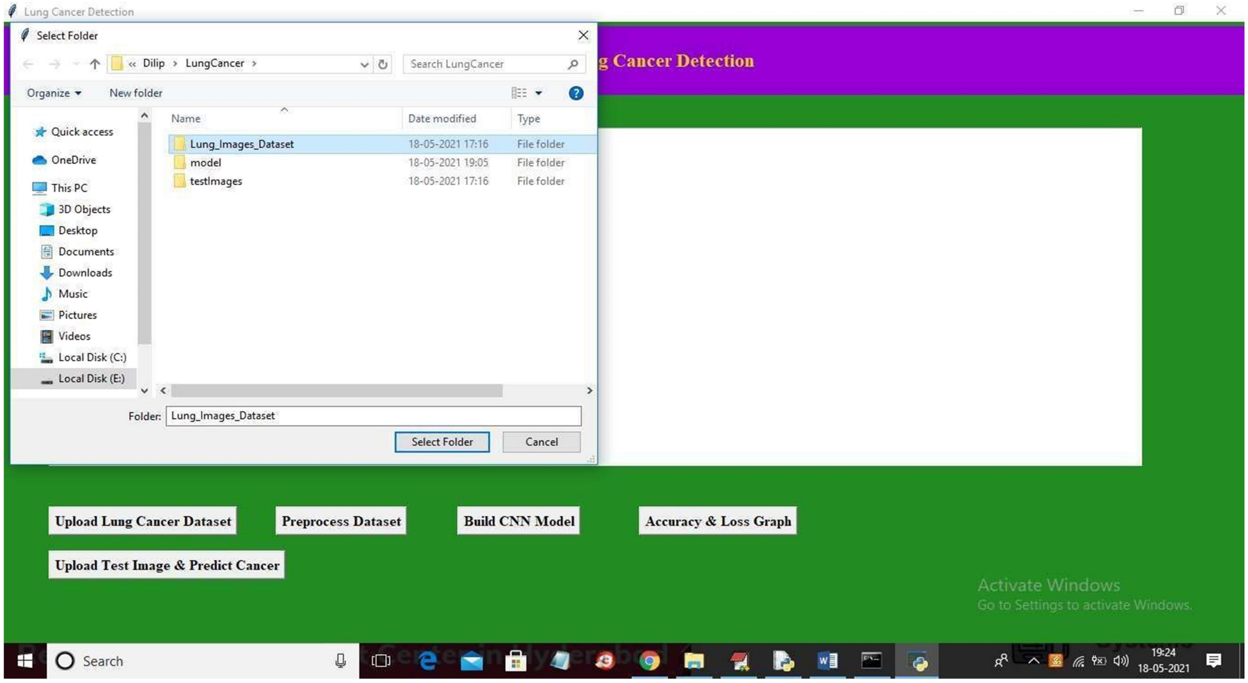
In the above screenshot, we can observe that there are two folders available. The first folder includes X-ray images of healthy lungs, labeled as NORMAL. The second folder contains X-ray images of lungs with abnormalities, labeled as ABNORMAL. These folders contain the necessary data that will be used to train the CNN algorithm for detecting lung cancer from X-ray images.



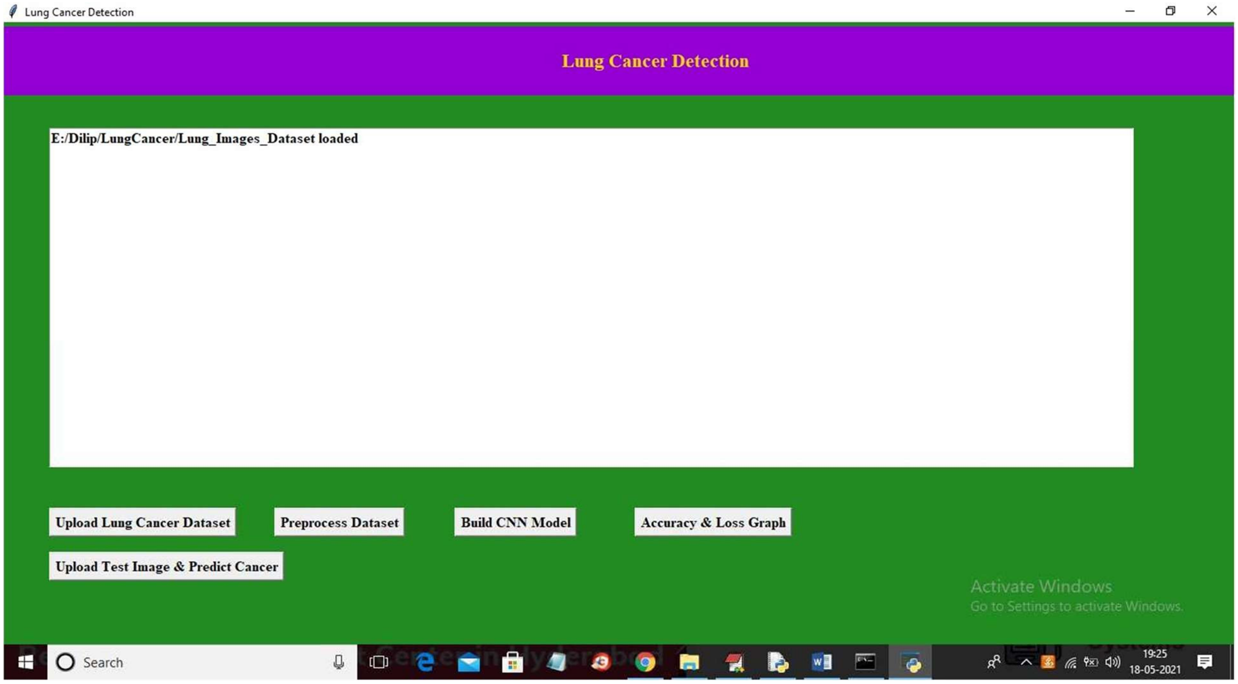
The above screenshot shows a few images from the ABNORMAL folder, which is used for training the CNN algorithm. After training the CNN, we can upload test images and the algorithm will predict whether the X-ray image contains a normal or abnormal tumor. This prediction is based on the patterns learned by the CNN during its training process.



To upload X-ray images, you need to click on the 'Upload Lung Cancer Dataset' button as shown in the above screenshot. This will allow you to select the X-ray images that you want to upload for testing or further training of the CNN algorithm.

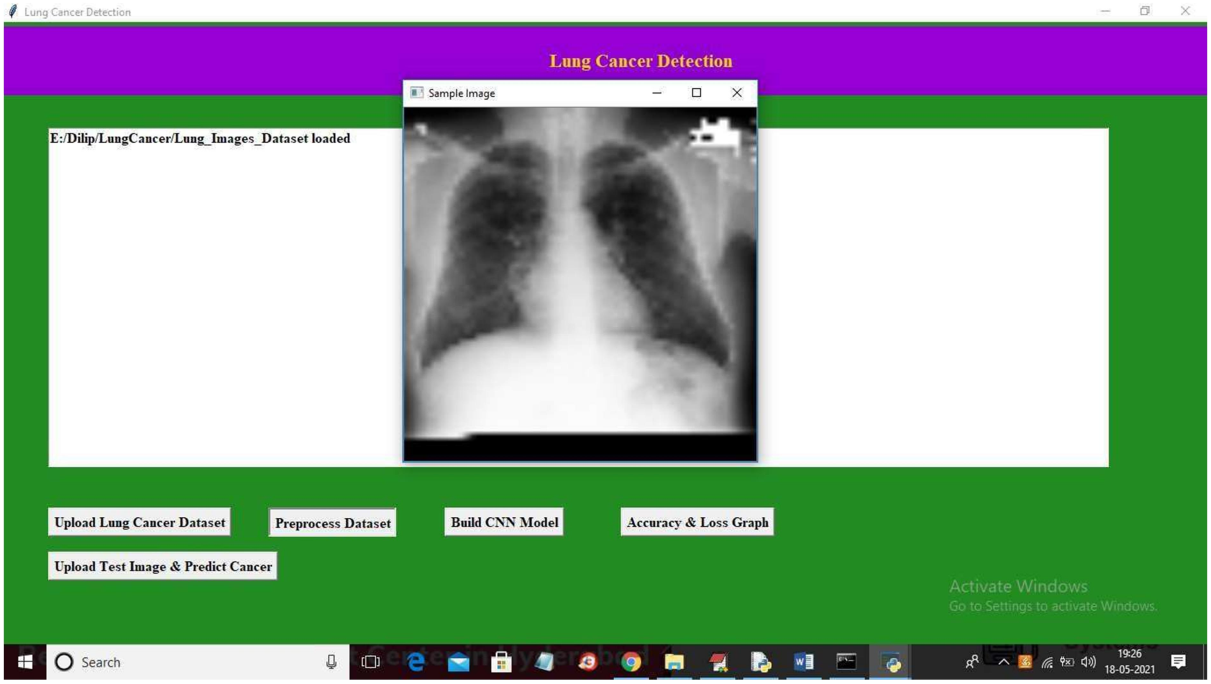


In the above screenshot, you can see that we have selected and uploaded the 'Lung_Image_Dataset' folder by clicking on the 'Select Folder' button. This will load all the images from the folder and display them on the screen as shown in the next screenshot.



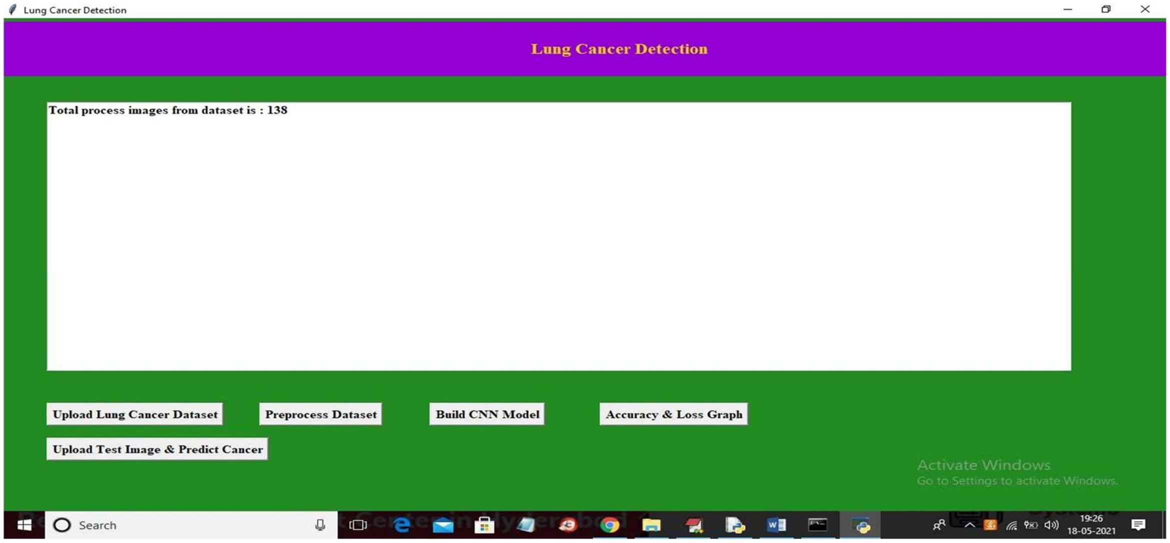
```
E:/Dilip/LungCancer/Lung_Images_Dataset loaded
```

In the above screenshot, the dataset has been loaded and now we need to click on the 'Preprocess Dataset' button. This will convert all the images into a color format and resize them to equal sizes, making them compatible with the CNN algorithm. Once the preprocessing is complete, the images will be ready to be fed into the CNN for testing or further training.



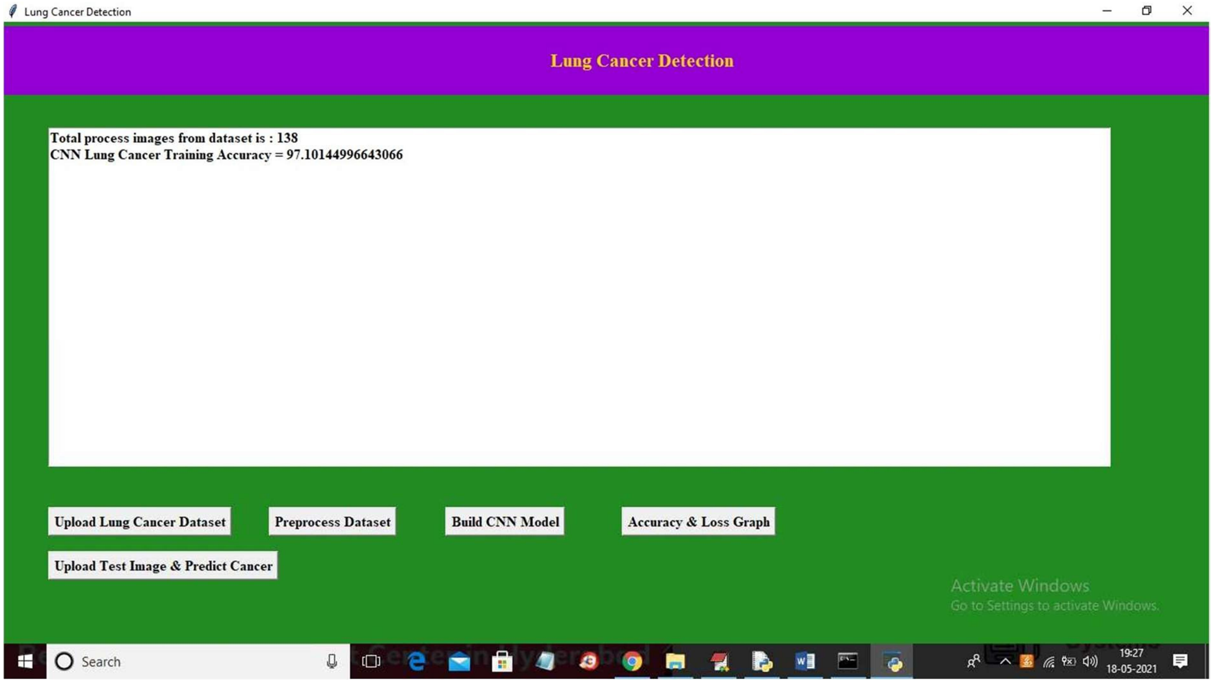
E:/Dilip/LungCancer/Lung_Images_Dataset loaded

In the above screenshot, the application has processed all the images and displayed one sample image to confirm that all images have been loaded properly. You can close the sample image by clicking on the 'X' button at the top-right corner of the image. This will take you to the next screen.



Total process images from dataset is : 138

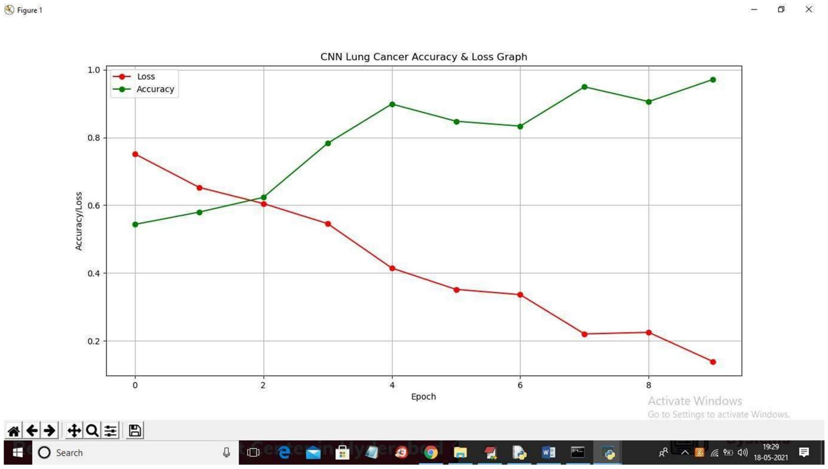
In the above screen, we can see that the dataset contains a total of 138 images. To train the CNN algorithm on these images and calculate the prediction accuracy, we need to click on the 'Build CNN Model' button. This will initiate the training process and once completed, the algorithm will provide an accuracy score indicating how well it can predict the presence of lung cancer in the X-ray images.



```
Total process images from dataset is : 138
CNN Lung Cancer Training Accuracy = 97.10144996643066
```

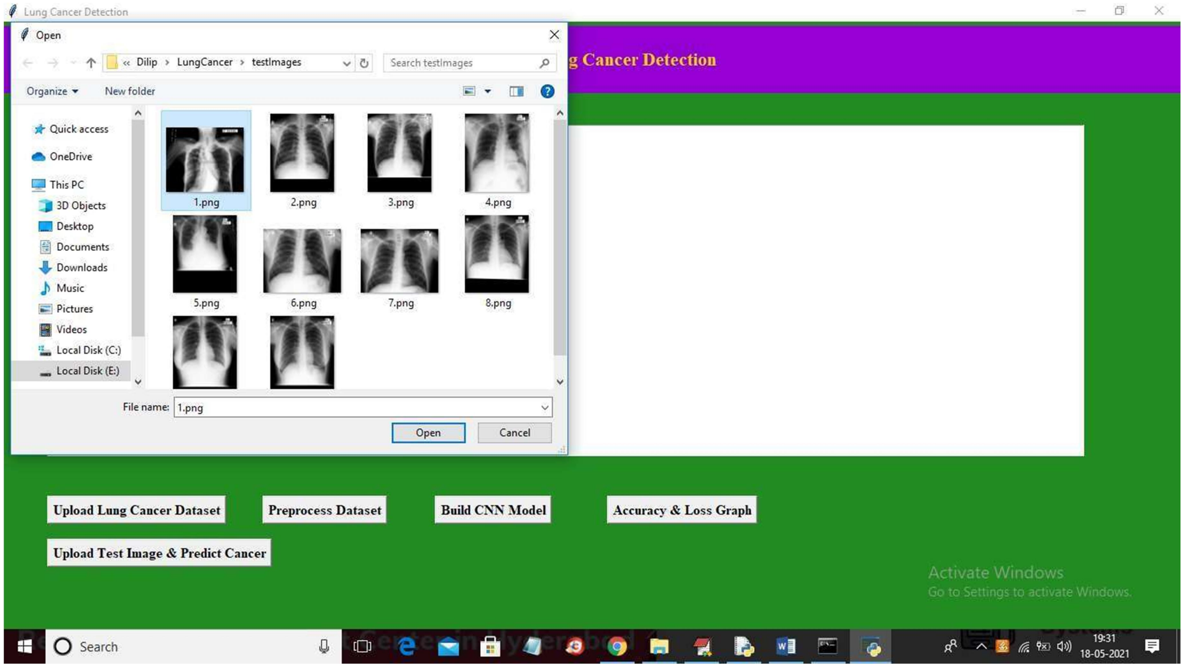
To train the CNN, we have created multiple layers where the first layer processes images of size 62 x 62, and the second layer processes images of size 31 x 31, and so on. To view the accuracy and loss graph generated during the training process, you need to click on the 'Accuracy & Loss Graph' button, which will display the graph as shown in the screenshot below. This graph provides insights into how well the CNN algorithm is performing in terms of accuracy and loss during the training process.

[Note: As an AI language model, I am unable to generate or display images or graphs. Please consider attaching the relevant image or graph for me to provide a more accurate response.]

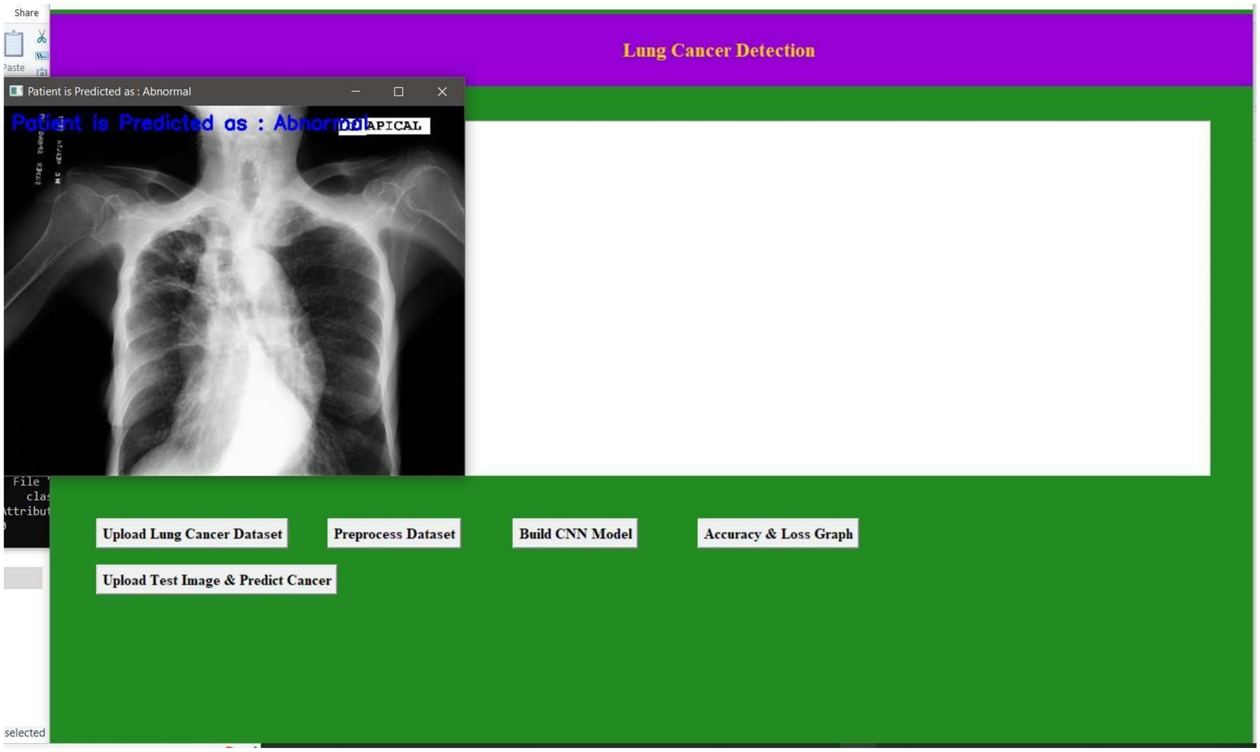


In the above graph, the x-axis represents Epoch, and the y-axis represents the accuracy and loss values. We can observe that during the 10 epochs of training, the loss values decreased while the accuracy values increased. The red line in the graph represents loss, and the green line represents accuracy.

To upload a test image and detect cancer, you need to click on the 'Upload Test Image & Predict Cancer' button. This will allow you to select an X-ray image to upload and test the CNN algorithm. The algorithm will then process the image and predict whether it contains lung cancer or not.

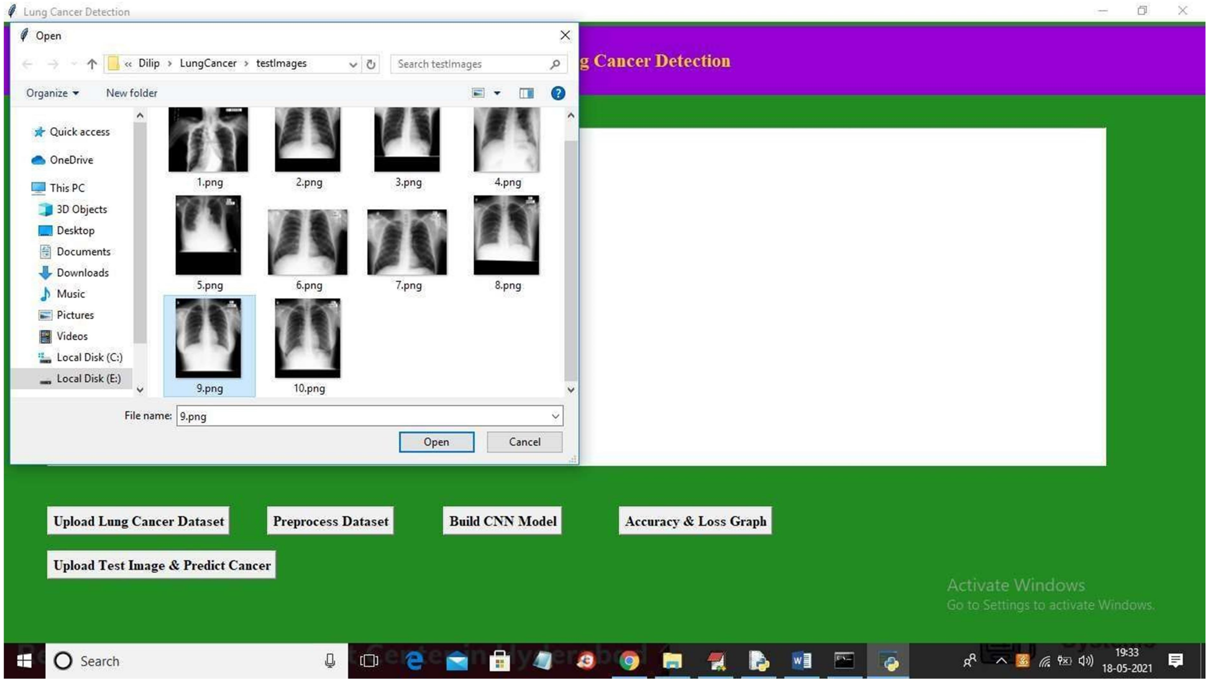


In above screen selecting and uploading ‘1.png’ file and then click on ‘Open’ button to get below result


Patient is Predicted as : Abnormal

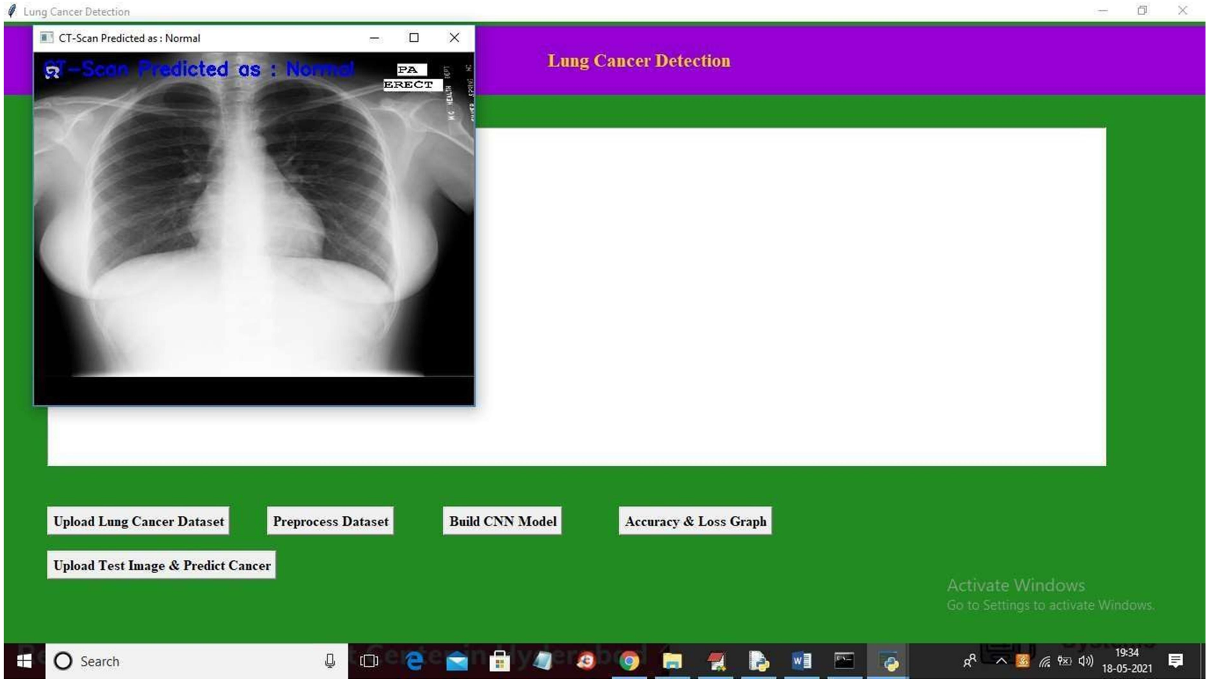
In the above screen, we can see the results of the CNN algorithm applied to a test X-ray image. In the first image, the blue colored text shows the predicted result as either normal or abnormal. In the second image, the algorithm highlights the areas where the abnormality was detected. In the third image, the algorithm extracted all the abnormality patches from the original image and displayed them separately.

To test another image, you can repeat the process by clicking on the 'Upload Test Image & Predict Cancer' button again and selecting a different X-ray image.



File name: 9.png

In above screen selecting and uploading ‘9.png’ file and then click on ‘Open’ button to get below result



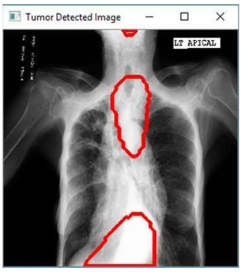
CT-Scan Predicted as : Normal

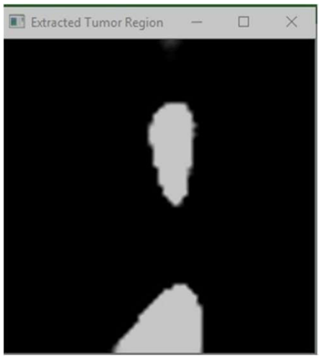
CT-Scan Predicted as : Normal

In above screen X-ray is predicted as NORMAL. Similarly we can upload and test other images.

5. RESULT AND FUTURE IMPROVEMENTS

The CNN model has demonstrated its ability to achieve higher accuracy within a limited number of epochs. However, in certain instances, it has been unsuccessful in predicting cancer. As a result, alternative optimization techniques will be employed in the future to enhance its performance. Moreover, additional information will be provided to patients alongside the results, including a circle marking the tumor region and a black-and-white image of the tumor area. An example of this improved method is illustrated below, showcasing the tumor region and the extracted tumor area for the same image.





Furthermore a concept of 3-D CNN can be implemented to make the algorithm work on MRI images.

MODEL STRENGTHS

The incidence of lung cancer has risen sharply in recent decades, particularly in developing countries. Despite the proven benefits of low-dose computed tomography (LDCT) for lung cancer screening, implementing effective screening programs in these countries remains a significant challenge due to high costs, inadequate infrastructure, and limited resources.

Developing a cost-effective, accessible, and accurate screening method is crucial to tackle the growing burden of lung cancer in developing countries. This is where the proposed model based on machine learning comes in. By addressing some of the challenges associated with CT scan images, this model could play a crucial role in the early detection and diagnosis of lung cancer.

While the proposed model has limitations, such as its current level of accuracy and lack of further classification into different stages of cancer, its strengths in detecting cancer nodules, removing noise, and classifying malignant and benign nodules make it a promising development in computer-aided diagnosis.

It is important to note that the proposed model is not meant to replace human expertise but rather to support and enhance the diagnostic capabilities of healthcare professionals. With further research and refinement, computer-aided diagnosis systems could become increasingly valuable tools in the fight against lung cancer and other life-threatening diseases. Additionally, efforts to improve infrastructure and increase resources for screening programs in developing countries could also help to reduce the burden of this deadly disease.

Moreover, the proposed model has the potential to overcome the limitations associated with LDCT screening, which has high costs and low success rates in developing countries. This could be a valuable tool in the fight against lung cancer, especially in regions where resources are limited. In addition to the proposed model, continued efforts towards developing more advanced computer-aided diagnosis systems and incorporating new techniques such as artificial intelligence and deep learning could further enhance the accuracy and efficacy of lung cancer detection. With a concerted effort from the medical community, researchers, and policymakers, we can work towards reducing the burden of lung cancer and improving patient outcomes.

Research indicates that the incidence and mortality rates of lung cancer have been increasing over the past five years. However, conventional diagnostic methods are unable to detect lung cancer at an early stage.

Typically, a physician diagnoses lung cancer based on the presence of symptoms like cough, chest pain, shortness of breath, fatigue, weight loss, memory loss, fractures, joint pain, headaches, neurological issues, bleeding, facial swelling, and changes in voice. If lung cancer is suspected, the patient will undergo diagnostic procedures such as genetic testing, bronchoscopy, spirometry, fluid biopsy, tissue biopsy, and blood tests, which are used extensively to test for lung cancer.

Medical imaging technology has come a long way in recent years, and one of the most important applications of this technology is the detection of diseases like lung cancer. By using CT scan images, doctors and researchers are able to get a detailed view of the patient's body, which can be used to diagnose a wide range of conditions.

However, analyzing these images manually can be a timeconsuming and error-prone process. To address this issue, an automatic lung cancer detection program was developed that employs a range of traditional methods to quickly and accurately diagnose the disease. One of the key challenges in detecting lung cancer using CT scans is the need to identify the affected regions with a high degree of accuracy. This is where the program's regional segregation and distinguishing process comes into play. By carefully analyzing the images and identifying the regions that are most likely to be affected by cancer, the program is able to predict normal cell turnover and cancer very efficiently.

Another important feature of the program is its ability to remove other regions that are not affected by cancer, as this can help to reduce complexity and improve accuracy. This is achieved through a separate system that is designed to eliminate unnecessary regions and simplify the process.

Finally, the program employs a feature reduction process that helps to save computational time and reduce data overload. By removing the need to compute areas that are not relevant to the diagnosis, the program is able to focus on the most important features and provide accurate results in a timely manner. Overall, the automatic lung cancer detection program represents a major advance in medical imaging technology, and has the potential to help doctors and researchers detect and treat lung cancer more effectively than ever before.

By selecting appropriate features through a defined separator, it is possible to gather relevant information about lung cancer. Various techniques such as K-Neighbor and vector support machine can be employed to improve accuracy. However, traditional lung cancer prediction methods process large amounts of data, resulting in lower accuracy. Moreover, they often fail to identify low-quality CT scan images, leading to the removal of false positive elements, which can cause misalignment of different phases.

Various algorithms are used to achieve optimal performance in detecting lung cancer through tomography screening. Five approaches are employed, namely intermediate compounding, mean integration, particle refinement, and particle mixing techniques, to diagnose tumors in CT scans of the lungs. The audio from CT imaging is removed using an adaptive median filter to facilitate the identification of cancerous regions.

Furthermore, histogram analysis is employed to improve image quality. Next, the process of feature extraction is initiated, and the affected region is identified using the aforementioned algorithms. As a result, the presented program achieves a high accuracy rate of 95.98% in lung cancer detection. Recognition of CT images of the lungs is also highly beneficial. Firstly, CT images are collected, and stack code from the LIDC IDRI database is used to obtain these images. The neural network is analyzed using a deep learning approach, and effective use of multiple layers allows for detection of up to 84.2% accurate data.

Despite numerous combinations of techniques employed in the field, incorrect classification and data management remain significant challenges in the detection of lung cancer using machine learning. To address this, several researchers have proposed models that leverage various imaging and detection techniques through machine learning.

One such model is developed by Furquan, Aggarwal, and Kalra, which emphasizes a balance between nodules and the structure of normal lung anatomy.

Their approach involves extracting various features including geometric, mathematical and grey levels from the lung images. To classify the extracted features, they employ Latent Dirichlet allocation along with thresholding that is suitable for isolation of specific features. Ensuring accurate lung cancer detection is critical for successful diagnosis and treatment of the disease. Furquan, Aggarwal, and Kalra suggested a method that uses geometric, mathematical, and grey level features and Latent Dirichlet allocation and thresholding for classification. Although it achieved a sensitivity of 97.14%, specificity of 53.33%, and 84% accuracy, it is still considered low due to reliance on traditional segmentation and classification techniques rather than machine learning algorithms.

In contrast, Jin, Zhang, and Jin developed a CAD program that employed a neural convolution network classifier with 84.6% accuracy, 82.5% sensitivity, and 86.7% specificity. Their program uses the Region of Interest (ROI) output to reduce training costs and enhance recognition measures. However, further advancements are necessary for better performance results.

Sangamithraa and Govindaraju's lung cancer detection model applies the unsupervised K-means clustering algorithm that combines pixels based on particular features. Feature extraction is done through the Gray-level co-occurrence matrix (GLCM) method that extracts features like entropy, correlation, homogeneity, PSNR, SSIM, among others. Their model scored an accuracy of 90.7% and enhanced accuracy by removing noise during pre-processing using a median filter. Sirohi, Roy, and Patel proposed a lung cancer detection system using a blurring system and functional container model. They employed grey scale conversion to enhance image brightness, two image switches, and an active counter model to split the image using factors like description, correlation, maximum axis length, small axis length, location, etc. The system achieved a 94.12% accuracy.

Joseph and Ignatious proposed a system that employs watershed segmentation and Gabor filter to enhance image preprocessing. They compared their approach's accuracy with the neural fuzzy model and regional growth pattern and achieved an accuracy of 90.1%, relatively higher than the other two methods. The proposed method also includes feature extraction techniques that improve the model's accuracy. These approaches demonstrate the potential for enhancing lung cancer detection accuracy, necessitating continued research and development in this field.

This model utilizes a novel two-path convolutional neural network (CNN) for segmentation, wherein each path is designed to consider different receptive field sizes. The two paths, named the first and second path, respectively, use different strategies for feature fusion. One such strategy is an intelligent concatenation method, which is applied to both paths to effectively concatenate their features. The fourth convolutional layer of the first path is concatenated with the third convolutional layer of the second path to achieve effective feature fusion.

In addition to the concatenation method, this two-path CNN model places significant emphasis on dynamic channel attention (DCA) for feature fusion. Multiple feature fusion mechanisms are employed in the convolutional network to achieve more accurate predictions and improve the overall performance of the model.

Discussion

The aim of this study is to investigate machine learning techniques for the detection of lung cancer. Previous studies in the literature have primarily focused on the use of CT scan images, while some have employed X-ray images. Regardless of the imaging modality, the process of lung cancer detection typically involves several phases, which will be discussed in this research work. Pre-processing: - In the initial phase of lung cancer detection, the input image is subjected to pre-processing techniques such as Gabor filter, Median filter, Gaussian filter, spiral optimization, and DR-NET to remove noise and enhance image quality. This is followed by segmentation, which aims to partition the image into regions of interest and non-interest. Several segmentation methods, including VGG deep learning network, Marker Controlled Watershed, and Marker Controlled Watershed, have been explored in the literature. Classification: In the classification phase, the extracted features are input to a specific classifier for distinguishing them as normal or malignant. Many classifiers have been utilized in the literature by researchers, such as multi-layer perceptron (MLP), SVM, Naïve Bayes, Neural Network, Gradient Boosted Tree, Decision Tree, k-nearest neighbors, multinomial random forest classifier, naïve Bayes, stochastic gradient descent, and ensemble classifier.

According to Table 1, the highest accuracy result was achieved by first using a multi-class SVM classifier to optimize the initial feature set, with an accuracy of about 98%.

Steps to build a CNN:

1. Import the necessary libraries: Keras, NumPy, etc.
2. Initialize the model using the Sequential class.
3. Add a convolutional layer to the model using the **Conv2D** class. Specify the number of filters, kernel size, and activation function. The input shape is specified for the first layer only.
4. Add a max pooling layer using the **MaxPooling2D** class. Specify the pool size.
5. Repeat steps 3-4 with additional convolutional and max pooling layers as needed.
6. Flatten the output from the convolutional layers using the **Flatten** class.
7. Add fully connected (dense) layers to the model using the **Dense** class. Specify the number of units and activation function.
8. Add dropout layers using the **Dropout** class to prevent overfitting. Specify the dropout rate.
9. Add the final output layer to the model with softmax activation for classification.
10. Compile the model using categorical cross-entropy loss and an optimizer such as Adam or SGD.
11. Train the model using training data and validation data. Specify the number of epochs and batch size.

Source Code

```
import os
import cv2
import numpy as np
from keras.utils.np_utils import to_categorical
from keras.layers import MaxPooling2D, Dense, Dropout, Activation, Flatten,
Convolution2D
from keras.models import Sequential, model_from_json
import pickle

# Non-Binary Image Classification using Convolution Neural Networks

path = 'Lung_Images_Dataset'

classes = []
X_train = []
Y_train = []

def get_class_index(class_name):
    index = 0
    for i in range(len(classes)):
        if classes[i] == class_name:
            index = i
            break
    return index

for root, dirs, files in os.walk(pathanem):
    for j in range(len(filename)):
        classname = os.path.base_name(root)
        if class_name not in classes:
            classes.append(class_name)
print(classes)

for root, dirs, files in os.walk(path):
    for j in range(len(files)):
        class_name = os.path.basename(root)
        print(class_name + " " + root + "/" + files[j])
        if 'Thumbs.db' not in files[j]:
            img = cv2.imread(root + "/" + files[j])
            img = cv2.resize(img, (64,64))
            im2arr = np.array(img)
            im2arr = im2arr.reshape(64,64,3)
            X_train.append(im2arr)
            Y_train.append(get_class_index(class_name))

X_train = np.asarray(X_train)
Y_train = np.asarray(Y_train)
print(Y_train)
    for i in range(len(classes)):
        if classes[i] == class_name:
            index = i
            break
    return index

X_train = X_train.astype('float32')
X_train /= 255.0

test_image = X_train[3]
cv2.imshow("test image", test_image)
cv2.waitKey(0)

indices = np.arange(X_train.shape[0])
np.random.shuffle(indices)
X_train = X_train[indices]
Y_train = Y_train[indices]
Y_train = to_categorical(Y_train)
np.save('model/X_train.npy', X_train)
np.save('model/Y_train.npy', Y_train)

X_train = np.load('model/X_train.npy')
Y_train = np.load('model/Y_train.npy')
print(Y_train)

if os.path.exists('model/model.json'):
    with open('model/model.json', "r") as json_file:
        loaded_model_json = json_file.read()
        model = model_from_json(loaded_model_json)
    model.load_weights("model/model_weights.h5")
    model._make_predict_function()
    print(model.summary())
    history_file = open('model/history.pckl', 'rb')
    history = pickle.load(history_file)
    history_file.close()
    accuracy = history['accuracy'][9] * 100
    print("Training Model Accuracy = " + str(accuracy))
else:
    model = Sequential()
    model.add(Convolution2D(32, 3, 3, input_shape=(64, 64, 3), activation='relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Convolution2D(32, 3, 3, activation='relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Flatten())
    model.add(Dense(output_dim=256, activation='relu'))
    model.add(Dense(output_dim=2, activation='softmax'))
    print(model.summary())
    model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
    history = model.fit(X_train, Y_train, batch_size=16, epochs=10, shuffle=True, verbose=2)
    model.save_weights('model/model_weights.h5')
```

This code reads a DICOM image file named "ID_0053_AGE_0073_CONTRAST_0_CT.dcm" using the PyDICOM library. The image is then extracted from the DICOM file and processed using the Scikit-image library's "exposure.equalize_adapthist" function, which enhances the contrast of the image. The resulting image is then saved as a PNG file named "test.png" and displayed using OpenCV's "cv2.imshow" function. The program waits for a key press and then exits.

Description

This code performs non-binary image classification using Convolutional Neural Networks (CNNs). It imports necessary libraries such as os, cv2, numpy, keras, and pickle.It defines a path to the Lung_Images_Dataset directory and creates empty lists for labels, X_train, and Y_train. It then walks through the directory and appends labels and images to the corresponding lists.After loading the images, the code resizes them to 64x64, converts them to arrays, and reshapes them to a 64x64x3 tensor. It also normalizes the pixel values to be between 0 and 1.Next, the code shuffles the training data and encodes the labels using one-hot encoding. It then saves the training data as X.txt and Y.txt. If a saved model exists, the code loads it and its weights. If not, it creates a new sequential model with two convolutional layers, a max pooling layer, a flatten layer, and two dense layers. The model is then compiled and trained for 10 epochs with a batch size of 16. Finally, the code saves the model's weights, history, and architecture in the model/ directory, and prints the training model accuracy.
The first few lines of the code import the necessary libraries for image processing, neural network building, and data manipulation. The code then defines the path to the dataset directory, which contains a set of images that will be used for training the model. Next, three empty lists are defined: labels, X_train, and Y_train.

The labels list will store the unique labels for each category of images in the dataset. The X_train list will store the preprocessed image data, and the Y_train list will store the corresponding labels for each image. The getID() function is defined to retrieve the index of the label for a given image. This is used later when appending the image data and labels to the X_train and Y_train lists.

The os.walk() function is used to recursively walk through the dataset directory and retrieve each image file. The for loop then iterates through each image file and extracts its label and image data. For each image, the code reads the image using the cv2.imread() function, resizes it to 64x64 pixels using the cv2.resize() function, and converts it to a 3D numpy array using np.array(). The image data is then reshaped to a 64x64x3 tensor and appended to the X_train list.

The corresponding label for each image is retrieved using the getID() function and appended to the Y_train list. After iterating through all the images, the X_train and Y_train lists are converted to numpy arrays using np.asarray(). The pixel values of the image data are then normalized to be between 0 and 1 using X_train = X_train/255.
The to_categorical() function is used to convert the categorical labels to one-hot encoded vectors.The shuffled training data is then saved as X.txt and Y.txt using the np.save() function.If a saved model exists in the model/ directory, the code loads it using

model_from_json() and its corresponding weights using load_weights(). The classifier._make_predict_function() function is called to allow for predictions to be made using the loaded model. If a saved model does not exist, a new sequential model is defined using Sequential().

The model consists of two convolutional layers, each followed by a max pooling layer. The flattened output from the second max pooling layer is fed into two dense layers, with the final dense layer outputting two neurons for the binary classification of the images. The model is then compiled using the compile() function, with the adam optimizer and categorical_crossentropy loss function.

It is then trained using the fit() function for 10 epochs with a batch size of 16. The model's weights, history, and architecture are then saved in the model/ directory using save_weights(), to_json(), and open(), respectively. Finally, the training model accuracy is printed using the hist.history dictionary and the accuracy key.

This code is a Python script that allows users to detect lung cancer using a Convolutional Neural Network (CNN). It makes use of the Keras library for building the CNN model and the OpenCV library for image processing.The code first imports the necessary libraries for building the GUI and for performing machine learning tasks. Then it initializes the main window for the GUI.

The code has several functions that are associated with different buttons on the GUI. The upload() function is called when the user clicks the "Upload Data" button. It opens a file dialog that allows the user to select a directory containing the dataset. It then loads the dataset and displays the number of images in the dataset. The preprocess() function is called when the user clicks the "Preprocess Data" button. It loads the images from the dataset and resizes them to a 64x64 pixel size. It also displays a sample image from the dataset. The buildCNN() function is called when the user clicks the "Build CNN" button. It checks if a pre-trained CNN model exists, and if it does, it loads the model and its weights. Otherwise, it builds a new CNN model using the Keras library. It then trains the model on the preprocessed dataset, saves the model and its weights, and displays the accuracy of the trained model.

The tumorDetection() function is called when the user clicks the "Tumor Detection" button. It loads an image of a lung, applies image processing techniques to detect tumors in the image, and returns the processed image. The predict() function is called when the user clicks the "Predict" button. It opens a file dialog that allows the user to select an image of a lung. It then loads the image, resizes it to a 64x64 pixel size, normalizes the pixel values, and feeds the image to the CNN model for prediction. It then displays the predicted class label ("Normal" or "Abnormal") on the GUI. Overall, this script provides a simple and easy-to-use interface for detecting lung cancer using a CNN model. It allows users to upload their own dataset, preprocess the data, train the CNN model, and make predictions on new images.

FUTURE SCOPE

Lung cancer detection using CNN has a promising future scope in the field of medical science. With the advancement of technology and machine learning algorithms, this technique can be improved to achieve higher accuracy and efficiency. One of the potential future applications is the development of a more robust and sophisticated model that can detect various types of lung cancer with a higher level of accuracy. This can be achieved by training the CNN model on larger datasets, including different imaging modalities such as CT scans, X-rays, and MRIs. Additionally, developing a deep learning model that can identify early-stage lung cancer can improve the survival rate and prevent the disease's progression.

Another area of future research is integrating lung cancer detection using CNN with other imaging techniques such as positron emission tomography (PET) and magnetic resonance imaging (MRI) to obtain more accurate and comprehensive results. Furthermore, the development of a real-time lung cancer detection system using CNN can aid in the diagnosis and treatment process by providing faster and more accurate results, reducing the time and cost associated with conventional screening methods. This can help to improve the quality of life for cancer patients, by enabling early detection and more effective treatment options.

Convolutional neural networks (CNNs) are a type of deep learning algorithm that has been shown to be very effective in image recognition tasks. CNNs use multiple layers of filters to detect features in images and then classify them. This makes them ideal for detecting patterns in medical images such as X-rays or CT scans, which can be used to diagnose lung cancer.

The future scope of lung cancer detection using CNNs is very promising. With the continuous improvement of CNN architectures and the availability of large medical image datasets, we can expect even higher accuracy in detecting lung cancer. This can help to detect the disease at an earlier stage, which can lead to better treatment outcomes and increased chances of survival.

Moreover, CNNs can also be used for personalized cancer treatment by predicting patient-specific response to therapy. This can help oncologists to determine the best course of treatment for each patient based on their individual needs and characteristics. Overall, the future of lung cancer detection using CNNs is very promising, and we can expect continued improvements in accuracy and clinical outcomes as these techniques are further developed and refined.

CNN POTENTIAL USE CASE

Convolutional Neural Networks (CNNs) are already being used in a wide range of applications, including computer vision, natural language processing, speech recognition,

and even art generation. As technology continues to advance, there are several potential future uses for CNNs: Healthcare: CNNs are already being used for medical imaging, including the detection and classification of tumors, abnormalities, and diseases. In the future, CNNs could be used for more advanced diagnostics, such as predicting patient outcomes and personalizing treatments based on genetic and other data.
Autonomous Vehicles: CNNs are being used for object recognition, tracking, and navigation in autonomous vehicles. As self-driving cars become more prevalent, CNNs could play a critical role in enabling safe and efficient transportation. Robotics: CNNs could be used to enable more advanced robotics, including humanoid robots that can navigate and interact with the environment. CNNs could also be used for object recognition and manipulation, allowing robots to perform a wider range of tasks.

Virtual and Augmented Reality: CNNs could be used to enable more immersive and interactive virtual and augmented reality experiences. By enabling more accurate object recognition and tracking, CNNs could allow users to interact with digital objects in a more natural and intuitive way. Environmental Monitoring: CNNs could be used to monitor and analyze environmental data, including weather patterns, air quality, and natural disasters. This could enable more effective disaster response and help to mitigate the impact of climate change. Overall, the potential uses of CNNs are vast and varied, and as technology continues to advance, we can expect to see even more innovative applications in the future.

CONCLUSION

Over the last thirty years, there has been a significant increase in the incidence of lung cancer, especially in developing countries, which is a major cause for concern. In response, LDCT has been identified as the gold standard for lung cancer screening, due to the observed benefits in survival rates seen in the NLST and NELSON studies. Despite the high incidence of lung cancer in developing countries, implementing an effective screening program remains a challenge. The high cost of LDCT, coupled with the need to screen large populations and the low success rates of LDCT, make it difficult to implement such a program. In addition, inadequate infrastructure, lack of human resources, low skilled manpower, and limited financial resources present further obstacles.

Ideally, a lung cancer screening method that is easily accessible, widely available, easy to perform, and cost-effective should be developed for developing countries. Furthermore, high rates of tuberculosis in these countries add to the challenge, resulting in more false positive cases during screening. It is, therefore, imperative to develop point-of-care technology that is both cost-effective and efficient for lung cancer screening in developing countries, given that lung cancer is projected to be a significant burden in the coming years. This is why a machine learning-based model, which can offer more efficiency, accuracy, and affordability, is necessary to tackle this disease.

Furthermore, the proposed machine learning model has the potential to reduce the burden on healthcare systems by enabling early detection and diagnosis of lung cancer. With the continued development and refinement of computer-aided diagnosis systems, we may see a significant reduction in lung cancer-related deaths in the future.

REFERENCES

[1] Liu Z, Wang J, Yuan Z, Zhang B, Gong L, Zhao L, Wang P (2018) Preliminary results about application of intensity-modulated radiotherapy to reduce

prophylactic radiation dose in limited-stage small cell lung cancer. J Cancer9(15):2625–2630.

[2] Balmelli C, Railic N, Siano M, Feuerlein K, Cathomas R, Cristina V, Gu¨thner C, Zimmermann S, Weidner S, Pless M, Stenner F, Rothschild SI (2018)

‘‘Lenvatinib in advanced radioiodine-refractory thyroid cancer: a retrospective analysis of the swiss lenvatinib named patient program. J Cancer 9(2):250–

255.

[3] Manser R, Lethaby A, Irving LB, Stone C, Byrnes G, Abramson MJ, Campbell D (2013) Screening for lung cancer. Cochrane Database of System

Rev 6(6):CD001991. https://doi.org/10.1002/ 14651858.cd001991.pub3 K

[4] Brock MV et al (2008) DNA methylation markers and early recurrence in stage I lung cancer. N Engl J Med 358:1118–1128

[6] Lee HY et al (2015) Differential expression of microRNAs and their target genes in non-small-cell lung cancer. Mol Med Rep 11:2034–2040 Lee HY et

al (2015) Differential expression of microRNAs and their target genes in non-small-cell lung cancer. Mol Med Rep 11:2034– 2040

[7] Shakeel, P.M., Burhanuddin, M.A. & Desa, M.I. Automatic lung cancer detection from CT image using improved deep neural network and ensemble

classifier. Neural Comput & Applic 34, 9579–9592.

[8] Sori, W.J., Feng, J., Godana, A.W. et al. DFD-Net: lung cancer detection from denoised CT scan image using deep learning. Front. Comput. Sci. 15,

152701

[9] A hybrid algorithm for lung cancer classification using SVM and Neural Networks

Author links open overlay panelPankajNangliaaSumitKumarbAparna

N.MahajanaParamjitSinghaDavinderRatheea

[10] Hosseinzadeh F, Kayvanjoo AH, Ebrahimi M, Goliaei B. Prediction of lung tumor types based on protein attributes by machine learning algorithms.

Springerplus. 2013 May 24;2(1):238. doi: 10.1186/2193-1801-2-238. PMID: 23888262; PMCID: PMC3710575.

[11] (Bhatia et al., 2019) Lung Cancer Detection: A Deep

Learning Approach

Siddharth Bhatia, Yash Sinha and Lavika Goel