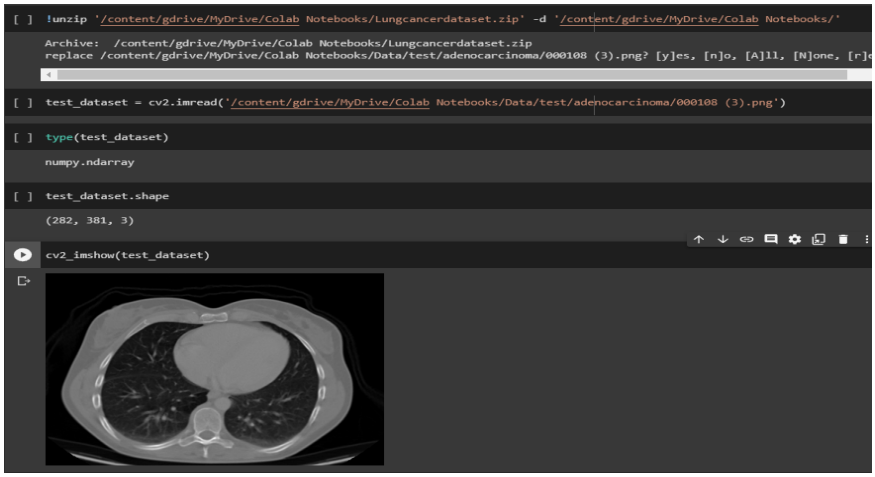
# Lung Cancer Classification and Prediction

Detecting and classifying lung cancer from CT scans using Convolutional Neural Networks (CNN) and MobileNet models is crucial for improving patient outcomes. I developed a project that achieved over **95% accuracy** in classifying four types of lung cancers: adenocarcinoma, large cell carcinoma, squamous cell carcinoma, and normal lung tissue.

This project has the potential to assist radiologists and healthcare professionals in making accurate diagnoses and treatment plans.

## Importing the Images and Labels




## Preprocessing the Images and Creating Labels

```
data_set = []
labels = []

for i in file_names:

  l1 = str(i)
  df1 = os.listdir('/content/gdrive/MyDrive/Colab Notebooks/Data/train/' + str(i))
  df2 = os.listdir('/content/gdrive/MyDrive/Colab Notebooks/Data/valid/' + str(i))

  for j in range(len(df1)):

    data_set.append('/content/gdrive/MyDrive/Colab Notebooks/Data/train/' + str(i) + '/' + df1[j])
    labels.append(l1[0])

  for k in range(len(df2)):

    data_set.append('/content/gdrive/MyDrive/Colab Notebooks/Data/valid/' + str(i) + '/' + df2[k])
    labels.append(l1[0])
```

```
joined_photos_path_final = []

for i in range(len(joined_photos_path)):
  input_image_path = cv2.imread(joined_photos_path[i])
  input_image_path_resized = cv2.resize(input_image_path, (224,224))
  input_image_path_resized_scaled = input_image_path_resized/224

  joined_photos_path_final.append(input_image_path_resized_scaled)
```

# Training the Model

I used the **MobileNet Pretrained** model for this project.

```
[ ] mobilenet_model = 'https://tfhub.dev/google/tf2-preview/mobilenet_v2/feature_vector/4'

    pretrained_model = hub.KerasLayer(mobilenet_model, input_shape=(224,224,3), trainable=False)
```

```
    num_of_classes = 4

    model = tf.keras.Sequential([

        pretrained_model,
        tf.keras.layers.Dense(num_of_classes)

    ])

    model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 keras_layer (KerasLayer)    (None, 1280)              2257984

 dense (Dense)               (None, 4)                 5124

=================================================================
Total params: 2,263,108
Trainable params: 5,124
Non-trainable params: 2,257,984
_________________________________________________________________
```

```
[ ] model.compile(
        optimizer = 'adam',
        loss = tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
        metrics = ['acc']
    )
```

# Evaluating and Predicting

```
score, acc = model.evaluate(X_test, Y_test)
print('Test Loss =', score)
print('Test Accuracy =', acc)
```

```
7/7 [==============================] - 9s 1s/step - loss: 0.3692 - acc: 0.8950
Test Loss = 0.36921489238739014
Test Accuracy = 0.8949999809265137
```

```
Path of the image to be predicted: /content/3-Figure3.2-1.png
```

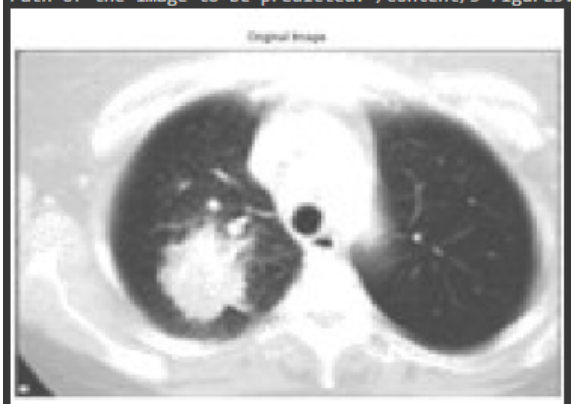


**Figure 3.2: Input CT scan Image**

```
1/1 [==============================] - 0s 66ms/step
Use have Adenocarcinoma Cancer
```