

Double-click (or enter) to edit

```
from google.colab import drive
drive.mount('/content/gdrive')
```

Mounted at /content/gdrive

```
import numpy as np
import pandas as pd
import os
for dirname, _, filenames in os.walk('/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

```
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/IQ-OTH_NCCD lung cancer dataset.txt
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (1).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (101).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (100).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (10).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (102).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (16).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (112).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (19).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (104).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (108).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (15).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (14).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (119).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (17).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (114).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (18).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (21).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (118).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (103).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (13).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (116).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (117).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (109).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (115).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (110).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (106).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (23).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (24).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (11).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (20).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (113).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (2).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (111).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (120).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (105).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (22).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (12).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (107).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (70).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (51).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (78).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (89).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (44).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (84).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (6).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (97).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (79).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (54).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (55).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (4).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (96).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (87).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (41).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (33).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (82).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (74).jpg
/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset/Bengin cases/Bengin case (48).jpg
```

```
import tensorflow as tf
from tensorflow.keras import models, layers
import matplotlib.pyplot as plt
from PIL import Image
import numpy as np
import pandas as pd
from tensorflow.keras.layers.experimental import preprocessing
from tensorflow.keras.preprocessing.image import load_img
```

```

import os
import PIL
import pathlib
import matplotlib.image as mpimg
import matplotlib.pyplot as plt
from tensorflow import keras
from tensorflow.keras import preprocessing
from tensorflow.keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten, Conv2D, MaxPool2D
from tensorflow.keras.optimizers import RMSprop
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, Flatten, Conv2D, MaxPooling2D
from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.python.ops.numpy_ops import np_utils

```

```

BATCH_SIZE = 62
IMAGE_SIZE = 256
EPOCHS=15
CHANNELS=3

```

```

dataset = tf.keras.preprocessing.image_dataset_from_directory(
    "/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset",
    seed=123,
    shuffle=True,
    image_size=(IMAGE_SIZE, IMAGE_SIZE),
    batch_size=BATCH_SIZE
)

```

Found 1097 files belonging to 3 classes.

```

class_names = dataset.class_names
class_names

```

['Benign cases', 'Malignant cases', 'Normal cases']

```
len(dataset)
```

18

```

for image_batch, label_batch in dataset.take(1):
    print(image_batch.shape)
    print(image_batch[1])
    print(label_batch.numpy())

```

```

(62, 256, 256, 3)
tf.Tensor(
[[[38. 38. 38.]
 [38. 38. 38.]
 [38. 38. 38.]
 ...
 [38. 38. 38.]
 [38. 38. 38.]
 [38. 38. 38.]]

[[38. 38. 38.]
 [38. 38. 38.]
 [38. 38. 38.]
 ...
 [38. 38. 38.]
 [38. 38. 38.]
 [38. 38. 38.]]

[[38. 38. 38.]
 [38. 38. 38.]
 [38. 38. 38.]
 ...
 [38. 38. 38.]
 [38. 38. 38.]
 [38. 38. 38.]]

...

[[38. 38. 38.]
 [38. 38. 38.]
 [38. 38. 38.]
 ...
 [38. 38. 38.]
 [38. 38. 38.]
 [38. 38. 38.]]

[[38. 38. 38.]
 [38. 38. 38.]
 [38. 38. 38.]

```

```

...
[38. 38. 38.]
[38. 38. 38.]
[38. 38. 38.]]

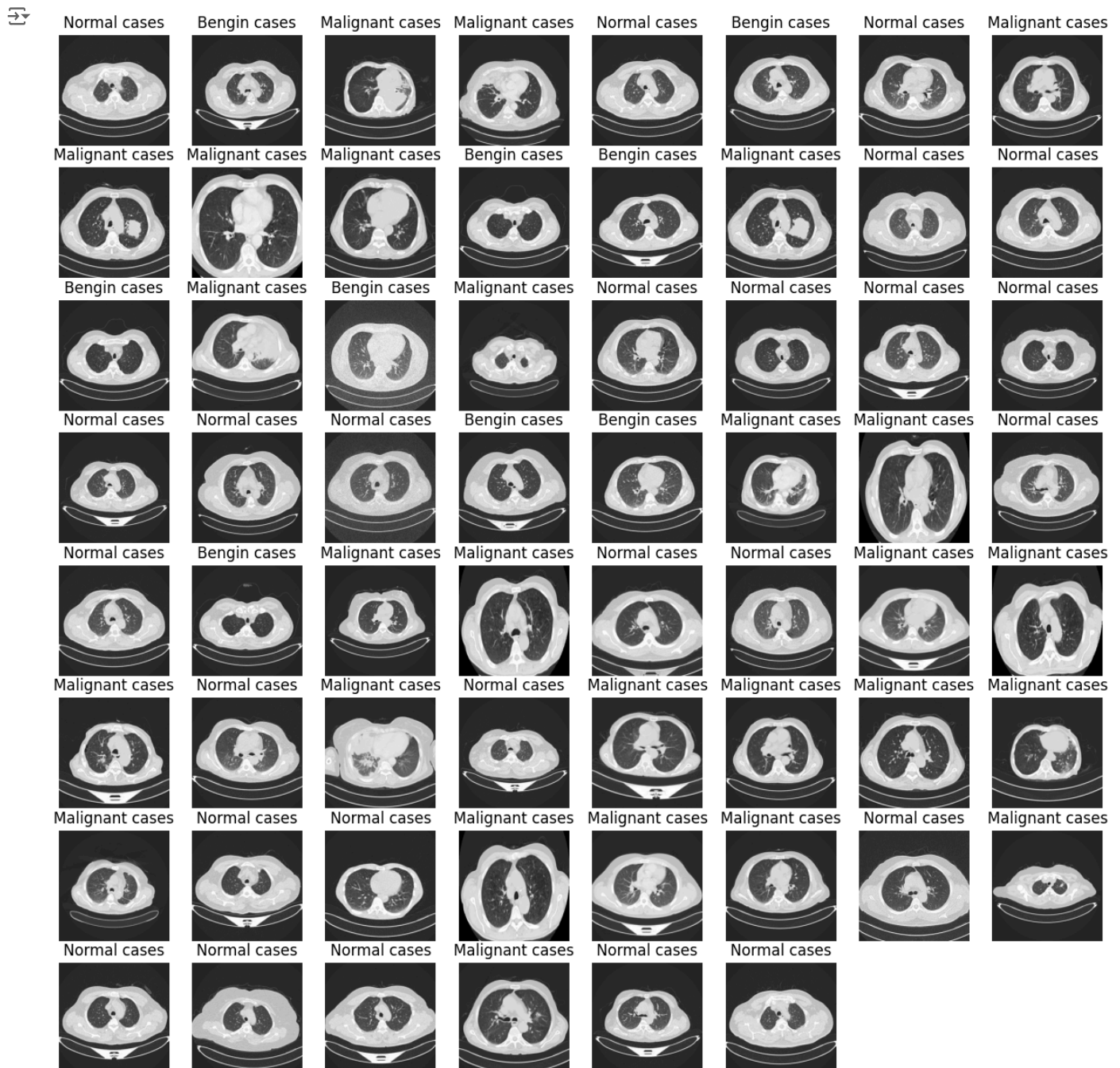
[[38. 38. 38.]
 [38. 38. 38.]
 [38. 38. 38.]]
...
[38. 38. 38.]
[38. 38. 38.]
[38. 38. 38.]]], shape=(256, 256, 3), dtype=float32)
[0 1 2 1 1 1 1 0 2 2 1 0 1 0 0 2 1 1 2 1 1 2 1 0 2 2 1 1 1 1 2 2 2 1 2 1 1
 2 1 1 2 2 1 2 1 2 0 2 2 2 1 0 1 1 2 0 1 1 0 0 2 1]

```

```

plt.figure(figsize=(15, 15))
for image_batch, labels_batch in dataset.take(1):
    for i in range(BATCH_SIZE):
        ax = plt.subplot(8, 8, i + 1)
        plt.imshow(image_batch[i].numpy().astype("uint8"))
        plt.title(class_names[labels_batch[i]])
        plt.axis("off")

```



```
def get_dataset_partitions_tf(ds, train_split=0.8, val_split=0.1, test_split=0.1, shuffle=True, shuffle_size=10000):
    assert (train_split + test_split + val_split) == 1
    ds_size = len(ds)
    if shuffle:
        ds = ds.shuffle(shuffle_size, seed=12)
    train_size = int(train_split * ds_size)
    val_size = int(val_split * ds_size)
    train_ds = ds.take(train_size)
    val_ds = ds.skip(train_size).take(val_size)
    test_ds = ds.skip(train_size).skip(val_size)
    # Autotune all the 3 datasets
    train_ds = train_ds.cache().shuffle(1000).prefetch(buffer_size=tf.data.AUTOTUNE)
    val_ds = val_ds.cache().shuffle(1000).prefetch(buffer_size=tf.data.AUTOTUNE)
    test_ds = test_ds.cache().shuffle(1000).prefetch(buffer_size=tf.data.AUTOTUNE)
    return train_ds, val_ds, test_ds
```

```
train_ds, val_ds, test_ds = get_dataset_partitions_tf(dataset)
```

```
resize_and_rescale = tf.keras.Sequential([
    layers.experimental.preprocessing.Resizing(IMAGE_SIZE, IMAGE_SIZE),
    layers.experimental.preprocessing.Rescaling(1./255),
])
```

```
data_augmentation = tf.keras.Sequential([
    layers.experimental.preprocessing.RandomFlip("horizontal_and_vertical"),
    layers.experimental.preprocessing.RandomRotation(0.2),
])
```

```
input_shape = (BATCH_SIZE, IMAGE_SIZE, IMAGE_SIZE, CHANNELS)
n_classes = 9
```

```
model = models.Sequential([
    resize_and_rescale,
    # data_augmentation,
    layers.Conv2D(32, kernel_size = (3,3), activation='relu', input_shape=input_shape),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, kernel_size = (3,3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, kernel_size = (3,3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Flatten(),
    layers.Dense(64, activation='relu'),
    layers.Dense(n_classes, activation='softmax'),
])
model.build(input_shape=input_shape)
```

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

```
model.summary()
```

↗ Model: "sequential\_2"

Layer (type)	Output Shape	Param #
=====		
sequential (Sequential)	(62, 256, 256, 3)	0
conv2d (Conv2D)	(62, 254, 254, 32)	896
max_pooling2d (MaxPooling2D)	(62, 127, 127, 32)	0
conv2d_1 (Conv2D)	(62, 125, 125, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(62, 62, 62, 64)	0
conv2d_2 (Conv2D)	(62, 60, 60, 64)	36928
max_pooling2d_2 (MaxPooling2D)	(62, 30, 30, 64)	0

```

g2D)

conv2d_3 (Conv2D)          (62, 28, 28, 64)          36928

max_pooling2d_3 (MaxPoolin (62, 14, 14, 64)          0
g2D)

conv2d_4 (Conv2D)          (62, 12, 12, 64)          36928

max_pooling2d_4 (MaxPoolin (62, 6, 6, 64)          0
g2D)

conv2d_5 (Conv2D)          (62, 4, 4, 64)            36928

max_pooling2d_5 (MaxPoolin (62, 2, 2, 64)          0
g2D)

flatten (Flatten)          (62, 256)                 0

dense (Dense)              (62, 64)                 16448

dense_1 (Dense)            (62, 9)                  585

=====
Total params: 184137 (719.29 KB)
Trainable params: 184137 (719.29 KB)
Non-trainable params: 0 (0.00 Byte)

```

```

history = model.fit(
    train_ds,
    batch_size=BATCH_SIZE,
    validation_data=val_ds,
    verbose=1,
    epochs=EPOCHS,
)

```

```

Epoch 1/15
14/14 [=====] - 103s 7s/step - loss: 0.0023 - accuracy: 1.0000 - val_loss: 7.5860e-04 - val_accuracy: 1.0000
Epoch 2/15
14/14 [=====] - 104s 7s/step - loss: 9.6465e-04 - accuracy: 1.0000 - val_loss: 0.0029 - val_accuracy: 1.0000
Epoch 3/15
14/14 [=====] - 102s 7s/step - loss: 0.0011 - accuracy: 1.0000 - val_loss: 0.0044 - val_accuracy: 1.0000
Epoch 4/15
14/14 [=====] - 101s 7s/step - loss: 0.0067 - accuracy: 0.9976 - val_loss: 2.1143e-04 - val_accuracy: 1.0000
Epoch 5/15
14/14 [=====] - 101s 7s/step - loss: 0.0229 - accuracy: 0.9918 - val_loss: 2.3813e-04 - val_accuracy: 1.0000
Epoch 6/15
14/14 [=====] - 99s 7s/step - loss: 0.3088 - accuracy: 0.9164 - val_loss: 0.1982 - val_accuracy: 0.9516
Epoch 7/15
14/14 [=====] - 99s 7s/step - loss: 0.1529 - accuracy: 0.9458 - val_loss: 0.0810 - val_accuracy: 0.9677
Epoch 8/15
14/14 [=====] - 97s 7s/step - loss: 0.0536 - accuracy: 0.9906 - val_loss: 0.0310 - val_accuracy: 0.9839
Epoch 9/15
14/14 [=====] - 97s 7s/step - loss: 0.0328 - accuracy: 0.9918 - val_loss: 0.0605 - val_accuracy: 0.9839
Epoch 10/15
14/14 [=====] - 102s 7s/step - loss: 0.0478 - accuracy: 0.9882 - val_loss: 0.0577 - val_accuracy: 0.9677
Epoch 11/15
14/14 [=====] - 97s 7s/step - loss: 0.0533 - accuracy: 0.9812 - val_loss: 0.0173 - val_accuracy: 1.0000
Epoch 12/15
14/14 [=====] - 97s 7s/step - loss: 0.0126 - accuracy: 0.9965 - val_loss: 0.0148 - val_accuracy: 1.0000
Epoch 13/15
14/14 [=====] - 99s 7s/step - loss: 0.0091 - accuracy: 0.9976 - val_loss: 0.0092 - val_accuracy: 1.0000
Epoch 14/15
14/14 [=====] - 97s 7s/step - loss: 0.0033 - accuracy: 0.9988 - val_loss: 0.0011 - val_accuracy: 1.0000
Epoch 15/15
14/14 [=====] - 99s 7s/step - loss: 0.0048 - accuracy: 0.9988 - val_loss: 0.0301 - val_accuracy: 0.9839

```

```
model.evaluate(test_ds)
```

```

3/3 [=====] - 11s 2s/step - loss: 0.1254 - accuracy: 0.9731
[0.1253802627325058, 0.9731183052062988]

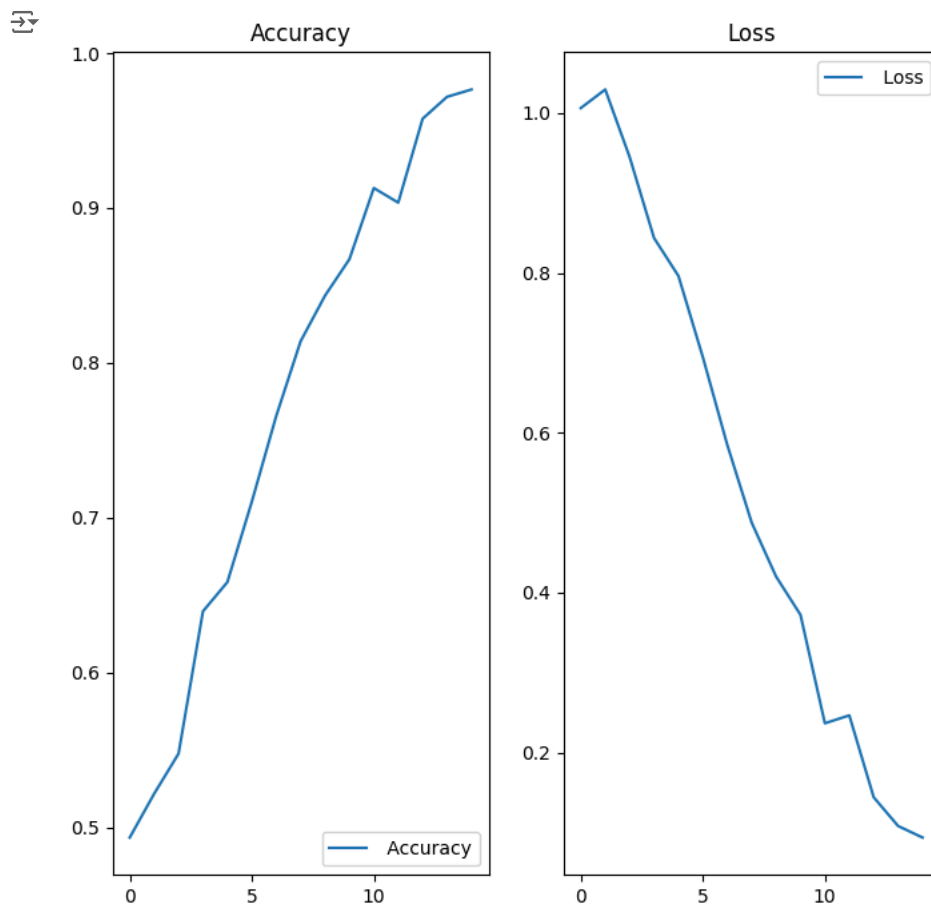
```

```

acc = history.history['accuracy']
loss = history.history['loss']

```

```
plt.figure(figsize=(8, 8))
plt.subplot(1, 2, 1)
plt.plot(range(EPOCHS), acc, label=' Accuracy')
plt.legend(loc='lower right')
plt.title('Accuracy')
plt.subplot(1, 2, 2)
plt.plot(range(EPOCHS), loss, label=' Loss')
plt.legend(loc='upper right')
plt.title('Loss')
plt.show()
```



```
image_path = "/content/drive/MyDrive/The IQ-OTHNCDD lung cancer dataset/Normal cases/Normal case (101).jpg"
image = preprocessing.image.load_img(image_path)
image_array = preprocessing.image.img_to_array(image)
scaled_img = np.expand_dims(image_array, axis=0)
image
```

