LUNG CANCER DETECTION USING YOLOV8 (Classification)

```
import os
import shutil
import random
from sklearn.model_selection import train_test_split

# Define paths to your folders
data_folders = ['Bengin cases', 'Malignant cases', 'Normal cases']
source_directory = r'/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset'
destination_directory = r'/content/drive/MyDrive/The IQ-OTHNCCD lung cancer dataset'

# Define ratios for train, validation, and test sets
train_ratio = 0.7
val_ratio = 0.15
test_ratio = 0.15


# Create destination directories for train, validation, and test sets
for folder in ['train', 'val', 'test']:
    for subfolder in data_folders:
        os.makedirs(os.path.join(destination_directory, folder, subfolder), exist_ok=True)

# Split data into train, validation, and test sets for each folder
for folder in data_folders:
    files = os.listdir(os.path.normpath(os.path.join(source_directory, folder)))
    random.shuffle(files)

    train_files, test_val_files = train_test_split(files, test_size=(val_ratio + test_ratio), random_state=42)
    val_files, test_files = train_test_split(test_val_files, test_size=test_ratio/(test_ratio + val_ratio), random_state=42)

    # Move files to their respective directories
    for file in train_files:
        shutil.move(os.path.join(source_directory, folder, file), os.path.join(destination_directory, 'train', folder))
    for file in val_files:
        shutil.move(os.path.join(source_directory, folder, file), os.path.join(destination_directory, 'val', folder))
    for file in test_files:
        shutil.move(os.path.join(source_directory, folder, file), os.path.join(destination_directory, 'test', folder))

print("Data split successfully.")
```

```
!nvidia-smi
```

```
Thu Mar  7 05:31:14 2024
+---------------------------------------------------------------------------------------+
| NVIDIA-SMI 535.104.05             Driver Version: 535.104.05   CUDA Version: 12.2     |
|-----------------------------------------+----------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |         Memory-Usage | GPU-Util  Compute M. |
|                                         |                      |               MIG M. |
|=========================================+======================+======================|
|   0  Tesla T4                       Off | 00000000:00:04.0 Off |                    0 |
| N/A   58C    P8              10W /  70W |      0MiB / 15360MiB |      0%      Default |
|                                         |                      |                  N/A |
+-----------------------------------------+----------------------+----------------------+

+---------------------------------------------------------------------------------------+
| Processes:                                                                            |
|  GPU   GI   CI        PID   Type   Process name                            GPU Memory |
|        ID   ID                                                             Usage      |
|=======================================================================================|
|  No running processes found                                                           |
+---------------------------------------------------------------------------------------+
```

```
import os
HOME = os.getcwd()
print(HOME)
```

```
/content
```

```
!pip install ultralytics==8.0.196

from IPython import display
display.clear_output()

import ultralytics
ultralytics.checks()
```

```
Ultralytics YOLOv8.0.196 🚀 Python-3.10.12 torch-2.1.0+cu121 CUDA:0 (Tesla T4, 15102MiB)
Setup complete ✅ (2 CPUs, 12.7 GB RAM, 26.4/78.2 GB disk)
```

```
from ultralytics import YOLO

from IPython.display import display, Image
```

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

```
Mounted at /content/drive
```

```
!yolo task=classify mode=train model='yolov8n-cls.pt' data='/content/drive/MyDrive/gdrive' epochs=100
```

```
     91/100     0.373G   0.001008          4        224: 100% 49/49 [00:09<00:00,  4.97it/s]
               classes   top1_acc   top5_acc: 100% 6/6 [00:01<00:00,  5.75it/s]
                   all      0.976          1

      Epoch    GPU_mem       loss  Instances       Size
     92/100     0.375G  0.0005679          4        224: 100% 49/49 [00:08<00:00,  5.49it/s]
               classes   top1_acc   top5_acc: 100% 6/6 [00:01<00:00,  5.79it/s]
                   all      0.976          1

      Epoch    GPU_mem       loss  Instances       Size
     93/100     0.375G  0.0007495          4        224: 100% 49/49 [00:07<00:00,  6.15it/s]
               classes   top1_acc   top5_acc: 100% 6/6 [00:01<00:00,  3.37it/s]
                   all      0.976          1

      Epoch    GPU_mem       loss  Instances       Size
     94/100     0.373G   0.007496          4        224: 100% 49/49 [00:07<00:00,  6.15it/s]
               classes   top1_acc   top5_acc: 100% 6/6 [00:01<00:00,  5.96it/s]
                   all      0.976          1

      Epoch    GPU_mem       loss  Instances       Size
     95/100     0.375G  0.0007422          4        224: 100% 49/49 [00:09<00:00,  5.35it/s]
               classes   top1_acc   top5_acc: 100% 6/6 [00:01<00:00,  5.69it/s]
                   all      0.976          1

      Epoch    GPU_mem       loss  Instances       Size
     96/100     0.375G   0.001092          4        224: 100% 49/49 [00:09<00:00,  5.28it/s]
               classes   top1_acc   top5_acc: 100% 6/6 [00:01<00:00,  5.84it/s]
                   all      0.976          1

      Epoch    GPU_mem       loss  Instances       Size
     97/100     0.373G  0.0009371          4        224: 100% 49/49 [00:07<00:00,  6.33it/s]
               classes   top1_acc   top5_acc: 100% 6/6 [00:01<00:00,  4.15it/s]
                   all      0.976          1

      Epoch    GPU_mem       loss  Instances       Size
     98/100     0.373G  0.0007337          4        224: 100% 49/49 [00:08<00:00,  5.72it/s]
               classes   top1_acc   top5_acc: 100% 6/6 [00:01<00:00,  5.64it/s]
                   all      0.976          1
Stopping training early as no improvement observed in last 50 epochs. Best results observed at epoch 48, best model saved as best
To update EarlyStopping(patience=50) pass a new patience value, i.e. `patience=300` or use `patience=0` to disable EarlyStopping.

98 epochs completed in 0.331 hours.
Optimizer stripped from runs/classify/train/weights/last.pt, 3.0MB
Optimizer stripped from runs/classify/train/weights/best.pt, 3.0MB

Validating runs/classify/train/weights/best.pt...
Ultralytics YOLOv8.0.196 🚀 Python-3.10.12 torch-2.1.0+cu121 CUDA:0 (Tesla T4, 15102MiB)
YOLOv8n-cls summary (fused): 73 layers, 1438723 parameters, 0 gradients, 3.3 GFLOPs
train: /content/drive/MyDrive/gdrive/train... found 772 images in 3 classes ✅
val: /content/drive/MyDrive/gdrive/val... found 165 images in 3 classes ✅
test: /content/drive/MyDrive/gdrive/test... found 168 images in 3 classes ✅
               classes   top1_acc   top5_acc: 100% 6/6 [00:01<00:00,  4.16it/s]
                   all      0.988          1
Speed: 0.1ms preprocess, 0.7ms inference, 0.0ms loss, 0.0ms postprocess per image
Results saved to runs/classify/train
Results saved to runs/classify/train
💡 Learn more at https://docs.ultralytics.com/modes/train
```

```
!yolo task=classify mode=val split=test batch=1 model=runs/classify/train/weights/best.pt data='/content/drive/MyDrive/gdrive'
```

```
Ultralytics YOLOv8.0.196 🚀 Python-3.10.12 torch-2.1.0+cu121 CUDA:0 (Tesla T4, 15102MiB)
YOLOv8n-cls summary (fused): 73 layers, 1438723 parameters, 0 gradients, 3.3 GFLOPs
train: /content/drive/MyDrive/gdrive/train... found 772 images in 3 classes ✅
val: /content/drive/MyDrive/gdrive/val... found 165 images in 3 classes ✅
test: /content/drive/MyDrive/gdrive/test... found 168 images in 3 classes ✅
test: Scanning /content/drive/MyDrive/gdrive/test... 168 images, 0 corrupt: 100% 168/168 [00:00<?, ?it/s]
               classes   top1_acc   top5_acc: 100% 168/168 [02:27<00:00,  1.14it/s]
                   all      0.982          1
Speed: 0.2ms preprocess, 3.7ms inference, 0.0ms loss, 0.0ms postprocess per image
Results saved to runs/classify/val
💡 Learn more at https://docs.ultralytics.com/modes/val
```

```
!yolo task=classify mode=predict save_txt=True model=runs/classify/train/weights/best.pt conf=0.25 save=True source=/content/Bengin-case
```

```
Ultralytics YOLOv8.0.196 🚀 Python-3.10.12 torch-2.1.0+cu121 CUDA:0 (Tesla T4, 15102MiB)
YOLOv8n-cls summary (fused): 73 layers, 1438723 parameters, 0 gradients, 3.3 GFLOPs

image 1/1 /content/Bengin-case-29-_jpg.rf.f9afac9e3fc492fc975e6bca98a9ebae.jpg: 224x224 Bengin cases 1.00, Normal cases 0.00, Malig
Speed: 1.6ms preprocess, 4.2ms inference, 0.1ms postprocess per image at shape (1, 3, 224, 224)
Results saved to runs/classify/predict
1 label saved to runs/classify/predict/labels
💡 Learn more at https://docs.ultralytics.com/modes/predict
```

```
with open('/content/runs/classify/predict/labels/Bengin-case-29-_jpg.rf.f9afac9e3fc492fc975e6bca98a9ebae.txt', 'r') as f:
    text = f.read()
print(text)
```

```
1.00 Bengin cases
0.00 Normal cases
0.00 Malignant cases
```

```
# Define the path to your text file
file_path = '/content/runs/classify/predict2/labels/Malignant-case-154-_jpg.rf.3cbfa978555e05727a27099627b41331.txt'

# Initialize an empty list to store the lines
lines = []

# Open the file and read its content
with open(file_path, 'r') as file:
    # Read each line and append it to the list
    for line in file:
        lines.append(line.strip())  # strip() removes leading and trailing whitespace
for i in lines:
  myString = i
  myList = myString.split(' ')
  length=len(myList)
  for j in range(length):
    if j==0:
      val=float(myList[j])
      if val>0.00:
        val_1=myList[j+1]
        if val_1=='Malignant':
          print('cancerous')
        elif val_1=='Bengin' or val_1=='Normal':
          print('Non cancerous')
    break
```

```
cancerous
```

```
# Define the path to your text file
file_path = '/content/runs/classify/predict3/labels/Normal-case-106-_jpg.rf.6d8694cf4f7f25734e421f073cad92d1.txt'

# Initialize an empty list to store the lines
lines = []

# Open the file and read its content
with open(file_path, 'r') as file:
    # Read each line and append it to the list
    for line in file:
        lines.append(line.strip())  # strip() removes leading and trailing whitespace
for i in lines:
  myString = i
  myList = myString.split(' ')
  length=len(myList)
  for j in range(length):
    if j==0:
      val=float(myList[j])
      if val>0.00:
        val_1=myList[j+1]
        if val_1=='Malignant':
          print('cancerous')
        elif val_1=='Bengin' or val_1=='Normal':
          print('Non cancerous')
    break
```

```
Non cancerous
```

```
# Define the path to your text file
file_path = '/content/runs/classify/predict/labels/Bengin-case-29-_jpg.rf.f9afac9e3fc492fc975e6bca98a9ebae.txt'

# Initialize an empty list to store the lines
lines = []

# Open the file and read its content
with open(file_path, 'r') as file:
    # Read each line and append it to the list
    for line in file:
        lines.append(line.strip())  # strip() removes leading and trailing whitespace
for i in lines:
  myString = i
  myList = myString.split(' ')
  length=len(myList)
  for j in range(length):
    if j==0:
      val=float(myList[j])
      if val>0.00:
        val_1=myList[j+1]
        if val_1=='Malignant':
          print('cancerous')
        elif val_1=='Bengin' or val_1=='Normal':
          print('Non cancerous')
    break
```

```
     Non cancerous
```

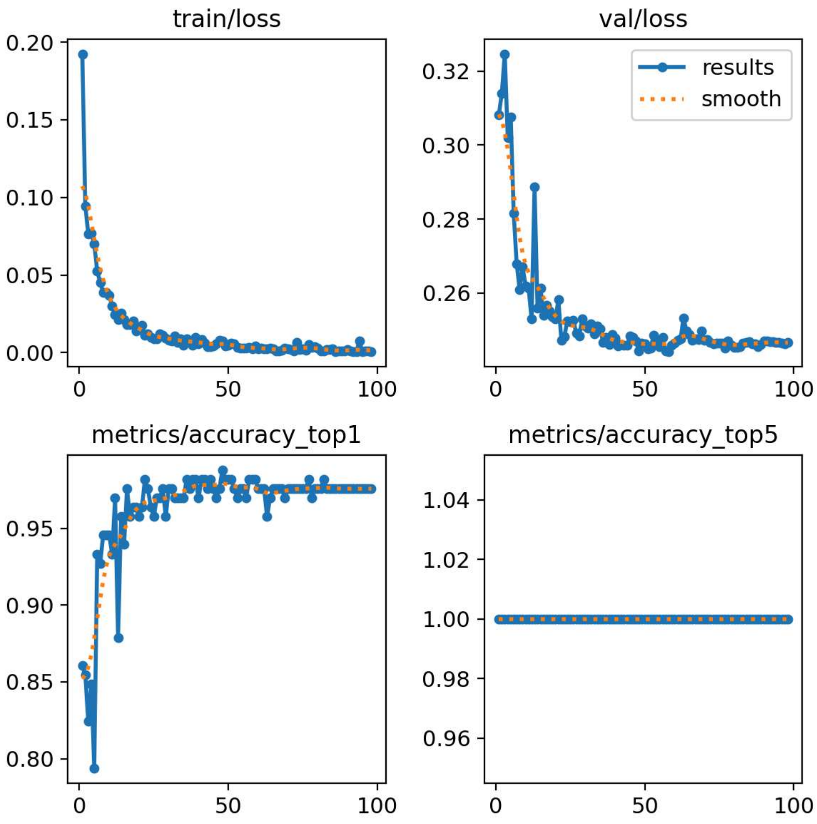
```
!yolo task=classify mode=predict save_txt=True show_labels=True model=runs/classify/train/weights/best.pt conf=0.25 save=True source=/c
```

```
     Ultralytics YOLOv8.0.196 🚀 Python-3.10.12 torch-2.1.0+cu121 CUDA:0 (Tesla T4, 15102MiB)
     YOLOv8n-cls summary (fused): 73 layers, 1438723 parameters, 0 gradients, 3.3 GFLOPs

     image 1/1 /content/Malignant-case-154-_jpg.rf.3cbfa978555e05727a27099627b41331.jpg: 224x224 Malignant cases 1.00, Bengin cases 0.00,
     Speed: 1.7ms preprocess, 4.7ms inference, 0.1ms postprocess per image at shape (1, 3, 224, 224)
     Results saved to runs/classify/predict4
     1 label saved to runs/classify/predict4/labels
     💡 Learn more at https://docs.ultralytics.com/modes/predict
```

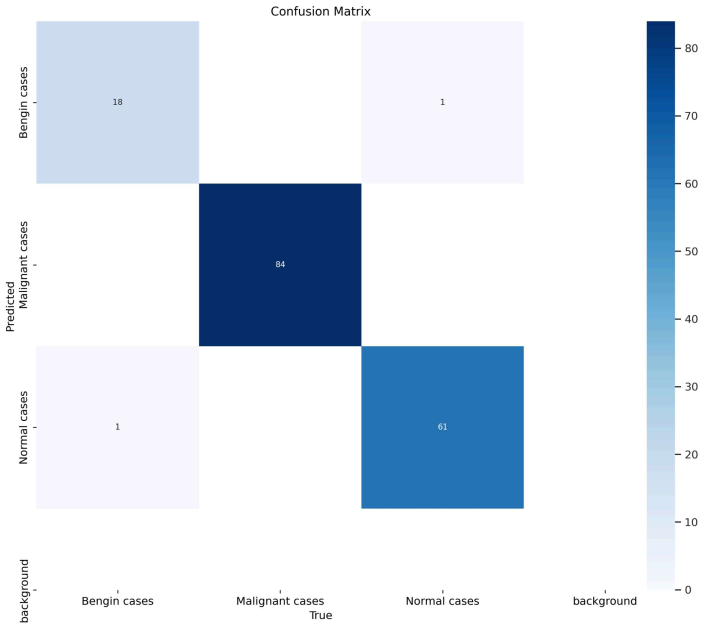
```
!yolo task=classify mode=predict save_txt=True model=runs/classify/train/weights/best.pt conf=0.25 save=True source=/content/Normal-case
```

```
     Ultralytics YOLOv8.0.196 🚀 Python-3.10.12 torch-2.1.0+cu121 CUDA:0 (Tesla T4, 15102MiB)
     YOLOv8n-cls summary (fused): 73 layers, 1438723 parameters, 0 gradients, 3.3 GFLOPs

     image 1/1 /content/Normal-case-106-_jpg.rf.6d8694cf4f7f25734e421f073cad92d1.jpg: 224x224 Normal cases 1.00, Malignant cases 0.00, Be
     Speed: 1.7ms preprocess, 4.4ms inference, 0.1ms postprocess per image at shape (1, 3, 224, 224)
     Results saved to runs/classify/predict3
     1 label saved to runs/classify/predict3/labels
     💡 Learn more at https://docs.ultralytics.com/modes/predict
```

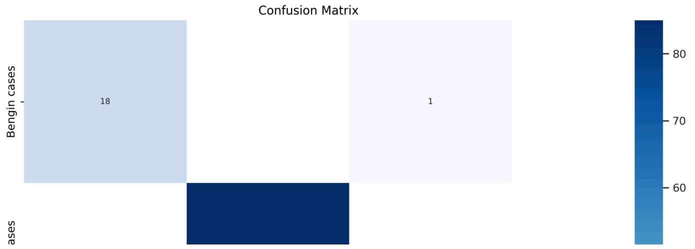
```
Image(filename =f'/content/runs/classify/train/results.png',width=640)
```



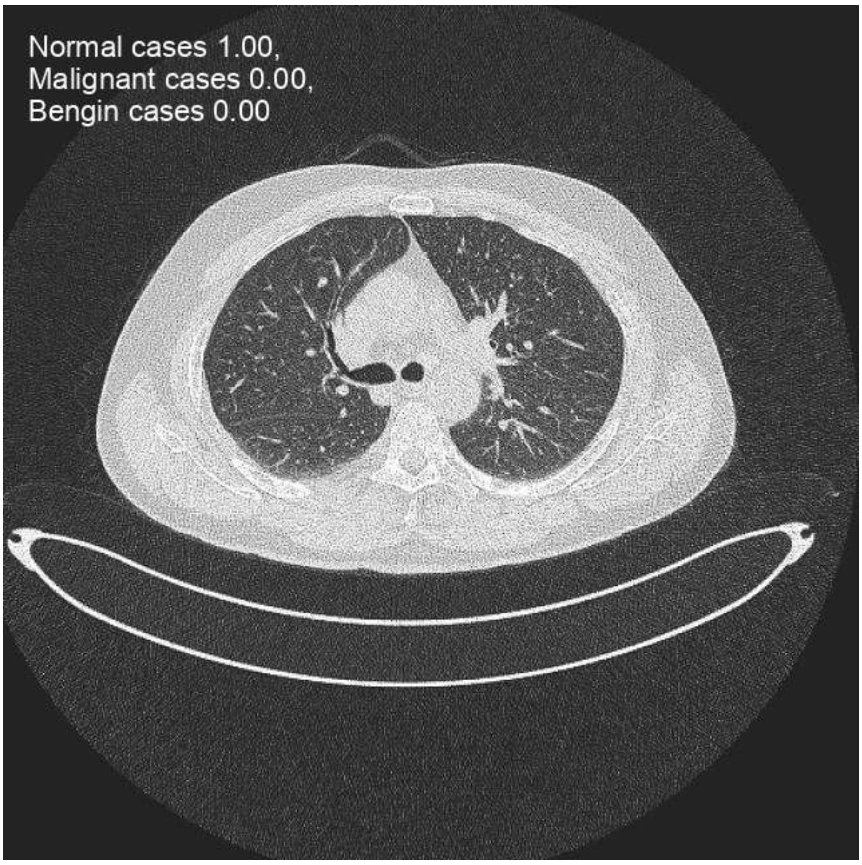
```
Image(filename =f'/content/runs/classify/train/confusion_matrix.png',width=640)
```



```
Image(filename =f'/content/runs/classify/val/confusion_matrix.png',width=640)
```



```
Image(filename =f'/content/runs/classify/predict3/Normal-case-106-_jpg.rf.6d8694cf4f7f25734e421f073cad92d1.jpg',width=640)
```



```
Image(filename =f'/content/runs/classify/predict/Bengin-case-29-_jpg.rf.f9afac9e3fc492fc975e6bca98a9ebae.jpg',width=640)
```