Pancreatic Adenocarcinoma Detection

About the model

The aim of this pancreatic adenocarcinoma detection is to develop an effective deep-learning based solution for identifying pancreatic cancer in abdomen in the CT – scan's of an individuals. The project utilizes the Ultralytics YOLO (You Only Look Once) model, a state-of-the-art real-time object detection algorithm, to accurately detect and localize **Malignant tumors** in CT scans. YOLO's speed and accuracy make it an ideal choice for real-time applications like this one.

The project involves setting up the environment, training the YOLO model on a custom dataset, evaluating its performance, and making predictions on test images. The ultimate goal is to enhance security and safety by automating the early stage detection of Malignant tumors in patients.

This command checks whether a compatible NVIDIA GPU is available for accelerated deep learning model training. GPUs are essential for efficiently training deep neural networks.

```
[2]  !nvidia-smi
```

```
+---------------------------------------------------------------------------------------+
| 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   36C    P8               9W /  70W |      0MiB / 15360MiB |      0%      Default |
|                                         |                      |                  N/A |
+-----------------------------------------+----------------------+----------------------+

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

- Setting Up the Environment

Here we get the current working directory and store it in the HOME variable. We then print the current working directory.

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

0s

- Installing Ultralytics

Here, we install the Ultralytics library, which is a popular framework for working with YOLO computer vision models. It provides tools and utilities for training and deploying object detection models.

```
!pip install ultralytics
Requirement already satisfied: numpy>=1.20 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.3.0->ultralytics) (1.25.2)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.3.0->ultralytics) (24.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.3.0->ultralytics) (3.1.2)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.3.0->ultralytics) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.1.4->ultralytics) (2023.4)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests>=2.23.0->ultralytics) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests>=2.23.0->ultralytics) (3.6)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests>=2.23.0->ultralytics) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests>=2.23.0->ultralytics) (2024.2.2)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from torch>=1.8.0->ultralytics) (3.13.3)
Requirement already satisfied: typing-extensions>=4.8.0 in /usr/local/lib/python3.10/dist-packages (from torch>=1.8.0->ultralytics) (4.10.0)
Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages (from torch>=1.8.0->ultralytics) (1.12)
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from torch>=1.8.0->ultralytics) (3.2.1)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from torch>=1.8.0->ultralytics) (3.1.3)
Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packages (from torch>=1.8.0->ultralytics) (2023.6.0)
Collecting nvidia-cuda-nvrtc-cu12==12.1.105 (from torch>=1.8.0->ultralytics)
  Downloading nvidia_cuda_nvrtc_cu12-12.1.105-py3-none-manylinux1_x86_64.whl (23.7 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 23.7/23.7 MB 41.6 MB/s eta 0:00:00
Collecting nvidia-cuda-runtime-cu12==12.1.105 (from torch>=1.8.0->ultralytics)
  Downloading nvidia_cuda_runtime_cu12-12.1.105-py3-none-manylinux1_x86_64.whl (823 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 823.6/823.6 kB 60.7 MB/s eta 0:00:00
Collecting nvidia-cuda-cupti-cu12==12.1.105 (from torch>=1.8.0->ultralytics)
  Downloading nvidia_cuda_cupti_cu12-12.1.105-py3-none-manylinux1_x86_64.whl (14.1 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 14.1/14.1 MB 53.5 MB/s eta 0:00:00
Collecting nvidia-cudnn-cu12==8.9.2.26 (from torch>=1.8.0->ultralytics)
  Downloading nvidia_cudnn_cu12-8.9.2.26-py3-none-manylinux1_x86_64.whl (731.7 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 731.7/731.7 MB 2.1 MB/s eta 0:00:00
Collecting nvidia-cublas-cu12==12.1.3.1 (from torch>=1.8.0->ultralytics)
  Downloading nvidia_cublas_cu12-12.1.3.1-py3-none-manylinux1_x86_64.whl (410.6 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 410.6/410.6 MB 2.4 MB/s eta 0:00:00
Collecting nvidia-cufft-cu12==11.0.2.54 (from torch>=1.8.0->ultralytics)
  Downloading nvidia_cufft_cu12-11.0.2.54-py3-none-manylinux1_x86_64.whl (121.6 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 121.6/121.6 MB 7.5 MB/s eta 0:00:00
Collecting nvidia-curand-cu12==10.3.2.106 (from torch>=1.8.0->ultralytics)
  Downloading nvidia_curand_cu12-10.3.2.106-py3-none-manylinux1_x86_64.whl (56.5 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 56.5/56.5 MB 12.5 MB/s eta 0:00:00
Collecting nvidia-cusolver-cu12==11.4.5.107 (from torch>=1.8.0->ultralytics)
```

✓ 0s completed at 11:23 AM

```
  Downloading nvidia_nccl_cu12-2.19.3-py3-none-manylinux1_x86_64.whl (166.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 166.0/166.0 MB 5.3 MB/s eta 0:00:00
Collecting nvidia-nvtx-cu12==12.1.105 (from torch>=1.8.0->ultralytics)
  Downloading nvidia_nvtx_cu12-12.1.105-py3-none-manylinux1_x86_64.whl (99 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 99.1/99.1 kB 13.3 MB/s eta 0:00:00
Requirement already satisfied: triton==2.2.0 in /usr/local/lib/python3.10/dist-packages (from torch>=1.8.0->ultralytics) (2.2.0)
Collecting nvidia-nvjitlink-cu12 (from nvidia-cusolver-cu12==11.4.5.107->torch>=1.8.0->ultralytics)
  Downloading nvidia_nvjitlink_cu12-12.4.99-py3-none-manylinux2014_x86_64.whl (21.1 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 21.1/21.1 MB 14.5 MB/s eta 0:00:00
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotlib>=3.3.0->ultralytics) (1.16.0)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->torch>=1.8.0->ultralytics) (2.1.5)
Requirement already satisfied: mpmath>=0.19 in /usr/local/lib/python3.10/dist-packages (from sympy->torch>=1.8.0->ultralytics) (1.3.0)
Installing collected packages: nvidia-nvtx-cu12, nvidia-nvjitlink-cu12, nvidia-nccl-cu12, nvidia-curand-cu12, nvidia-cufft-cu12, nvidia-cuda-runtime-cu12, nvidia-c
Successfully installed nvidia-cublas-cu12-12.1.3.1 nvidia-cuda-cupti-cu12-12.1.105 nvidia-cuda-nvrtc-cu12-12.1.105 nvidia-cuda-runtime-cu12-12.1.105 nvidia-cudnn-c
```

- Importing and Checking Ultralytics

In this part, we import the Ultralytics library and use the clear_output function to clear the current output. Then, we run checks to ensure that Ultralytics is correctly installed and configured.

```
from IPython import display
display.clear_output()

import ultralytics
ultralytics.check()
```

```
Ultralytics YOLOv8.0.227 🚀 Python-3.10.12 torch-2.1.0+cu121 CPU (Intel Xeon 2.20GHz)
Setup complete ✅ (2 CPUs, 12.7 GB RAM, 26.2/107.7 GB disk)
```

```
[8] import ultralytics
    from ultralytics.utils.checks import check_pip_update_available

    # Check for updates
    update_available = check_pip_update_available()

    if update_available:
      print("An update for ultralytics is available!")
    else:
      print("ultralytics is up to date.")
```

```
ultralytics is up to date.
```

- Importing YOLO from Ultralytics & Mounting Google Drive

We import the YOLO class from Ultralytics. This class is used for creating and managing YOLO models for object detection tasks.

In the next part we mount Google Drive into the Colab environment. It's a common practice for accessing data and project files stored in your Google Drive. We then navigate to the directory where your project files are located.

```
1m  [1]  from google.colab import drive
         drive.mount('/content/drive')

         Mounted at /content/drive
```

- Train / Val / Test Split

The dataset consists of 120 images for the training set, 30 images for the validation set and 5 images for the test set.
Hence, the data consists of a split of approximately 77% images in the train set, 20% in the validation set and 3% in the test set.

- Model Training

In this step, we start training a YOLO model. The command specifies various parameters, such as the training mode, the YOLO model architecture (yolov8s), the data configuration file (configdata.yaml), the number of training epochs (20), the input image size (imgsz), and whether to generate training plots (plots=True).

```python
import os

from ultralytics import YOLO


# Load a model
model = YOLO("yolov8n.yaml")  # build a new model from scratch

# Use the model
results = model.train(data=os.path.join(ROOT_DIR, "configdata.yaml"), epochs=20)  # train the model
```

```
Ultralytics YOLOv8.1.37 🚀 Python-3.10.12 torch-2.2.1+cu121 CUDA:0 (Tesla T4, 15102MiB)
engine/trainer: task=detect, mode=train, model=yolov8n.yaml, data=/content/gdrive/My Drive/pan_can/configdata.yaml, epochs=2, time=None, patience=100, batch=16, imgsz=640, save=True, save_period=-1, ca
Downloading https://ultralytics.com/assets/Arial.ttf to '/root/.config/Ultralytics/Arial.ttf'...
100%|██████████| 755k/755k [00:00<00:00, 17.8MB/s]
Overriding model.yaml nc=80 with nc=1

                   from  n    params  module                                       arguments
  0                  -1  1       464  ultralytics.nn.modules.conv.Conv             [3, 16, 3, 2]
  1                  -1  1      4672  ultralytics.nn.modules.conv.Conv             [16, 32, 3, 2]
  2                  -1  1      7360  ultralytics.nn.modules.block.C2f             [32, 32, 1, True]
  3                  -1  1     18560  ultralytics.nn.modules.conv.Conv             [32, 64, 3, 2]
  4                  -1  2     49664  ultralytics.nn.modules.block.C2f             [64, 64, 2, True]
  5                  -1  1     73984  ultralytics.nn.modules.conv.Conv             [64, 128, 3, 2]
  6                  -1  2    197632  ultralytics.nn.modules.block.C2f             [128, 128, 2, True]
  7                  -1  1    295424  ultralytics.nn.modules.conv.Conv             [128, 256, 3, 2]
  8                  -1  1    460288  ultralytics.nn.modules.block.C2f             [256, 256, 1, True]
  9                  -1  1    164608  ultralytics.nn.modules.block.SPPF            [256, 256, 5]
 10                  -1  1         0  torch.nn.modules.upsampling.Upsample         [None, 2, 'nearest']
 11             [-1, 6]  1         0  ultralytics.nn.modules.conv.Concat           [1]
 12                  -1  1    148224  ultralytics.nn.modules.block.C2f             [384, 128, 1]
 13                  -1  1         0  torch.nn.modules.upsampling.Upsample         [None, 2, 'nearest']
 14             [-1, 4]  1         0  ultralytics.nn.modules.conv.Concat           [1]
 15                  -1  1     37248  ultralytics.nn.modules.block.C2f             [192, 64, 1]
 16                  -1  1     36992  ultralytics.nn.modules.conv.Conv             [64, 64, 3, 2]
 17            [-1, 12]  1         0  ultralytics.nn.modules.conv.Concat           [1]
 18                  -1  1    123648  ultralytics.nn.modules.block.C2f             [192, 128, 1]
 19                  -1  1    147712  ultralytics.nn.modules.conv.Conv             [128, 128, 3, 2]
 20             [-1, 9]  1         0  ultralytics.nn.modules.conv.Concat           [1]
 21                  -1  1    493056  ultralytics.nn.modules.block.C2f             [384, 256, 1]
 22        [15, 18, 21]  1    751507  ultralytics.nn.modules.head.Detect           [1, [64, 128, 256]]
YOLOv8n summary: 225 layers, 3011043 parameters, 3011027 gradients, 8.2 GFLOPs
```

```
      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       4/10      2.26G      3.133      4.959      4.062         11        640: 100%|██████████| 10/10 [00:02<00:00,  3.51it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 5/5 [00:01<00:00,  2.66it/s]                all

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       5/10      2.25G      3.124      4.848      3.998         13        640: 100%|██████████| 10/10 [00:03<00:00,  2.66it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 5/5 [00:02<00:00,  2.49it/s]                all

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       6/10      2.25G      2.889      4.671      3.892         11        640: 100%|██████████| 10/10 [00:02<00:00,  4.05it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 5/5 [00:01<00:00,  2.63it/s]                all

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       7/10      2.25G      2.951      4.711      3.795         12        640: 100%|██████████| 10/10 [00:02<00:00,  3.75it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 5/5 [00:03<00:00,  1.42it/s]                all

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       8/10      2.25G      2.925      4.531      3.635         12        640: 100%|██████████| 10/10 [00:03<00:00,  2.79it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 5/5 [00:03<00:00,  1.30it/s]                all

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       9/10      2.25G      2.812      4.533      3.551         12        640: 100%|██████████| 10/10 [00:04<00:00,  2.38it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 5/5 [00:02<00:00,  2.13it/s]                all
```

```
Optimizer stripped from runs/detect/train2/weights/last.pt, 6.2MB
Optimizer stripped from runs/detect/train2/weights/best.pt, 6.2MB

Validating runs/detect/train2/weights/best.pt...
Ultralytics YOLOv8.1.37 🚀 Python-3.10.12 torch-2.2.1+cu121 CUDA:0 (Tesla T4, 15102MiB)
YOLOv8n summary (fused): 168 layers, 3005843 parameters, 0 gradients, 8.1 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 5/5 [00:05<00:00,  1.15s/it]
                   all        155        170   0.000436      0.118   0.000304   7.34e-05
Speed: 0.6ms preprocess, 4.9ms inference, 0.0ms loss, 7.5ms postprocess per image
Results saved to runs/detect/train2
```

- Data Augmentation
- **optimizer:** 'optimizer=auto' found, ignoring 'lr0=0.01' and 'momentum=0.937' and determining best 'optimizer', 'lr0' and 'momentum' automatically...
- **optimizer:** AdamW(lr=0.002, momentum=0.9) with parameter groups 57 weight(decay=0.0), 64 weight(decay=0.0005), 63 bias(decay=0.0)

```
optimizer: 'optimizer=auto' found, ignoring 'lr0=0.01' and 'momentum=0.937' and determining best 'optimizer', 'lr0' and 'momentum' automatically...
optimizer: AdamW(lr=0.002, momentum=0.9) with parameter groups 57 weight(decay=0.0), 64 weight(decay=0.0005), 63 bias(decay=0.0)
```

```
albumentations: Blur(p=0.01, blur_limit=(3, 7)), MedianBlur(p=0.01, blur_limit=(3, 7)), ToGray(p=0.01), CLAHE(p=0.01, clip_limit=(1, 4.0), tile_grid_size=(8, 8))
```

These are augmentation operations defined using the Albumentations library. Albumentations is a library for image augmentation, used to increase the diversity of training data.

Blur(p=0.01, blur_limit=(3, 7)):
This augmentation applies blur to the image with a probability of 0.01. The blur_limit parameter specifies the range of blur strength, which is a random value between 3 and 7. Blur is a common image processing operation that reduces image noise and detail.

MedianBlur(p=0.01, blur_limit=(3, 7)):
This augmentation applies median blur to the image with a probability of 0.01. Similar to the Blur operation, the blur_limit parameter specifies the range of blur strength.

ToGray(p=0.01):
This augmentation converts the image to grayscale with a probability of 0.01. Grayscale images contain only shades of gray (black, white, and various gray levels), removing color information from the original image.

CLAHE(p=0.01, clip_limit=(1, 4.0), tile_grid_size=(8, 8)): This augmentation applies Contrast Limited Adaptive Histogram Equalization (CLAHE) to the image with a probability of 0.01. CLAHE enhances the image contrast by redistributing pixel intensities based on local histogram equalization. The clip_limit parameter controls the contrast, and it's a random value between 1 and 4.0. The tile_grid_size parameter specifies the size of the grid for histogram equalization.

These augmentations are applied to input images during the training phase to create variations of the original data. By applying random transformations to the images, the model is exposed to a broader range of scenarios, helping it generalize better to different conditions. The low probabilities (e.g., 0.01) indicate that these augmentations are applied infrequently to avoid overdoing the transformations.

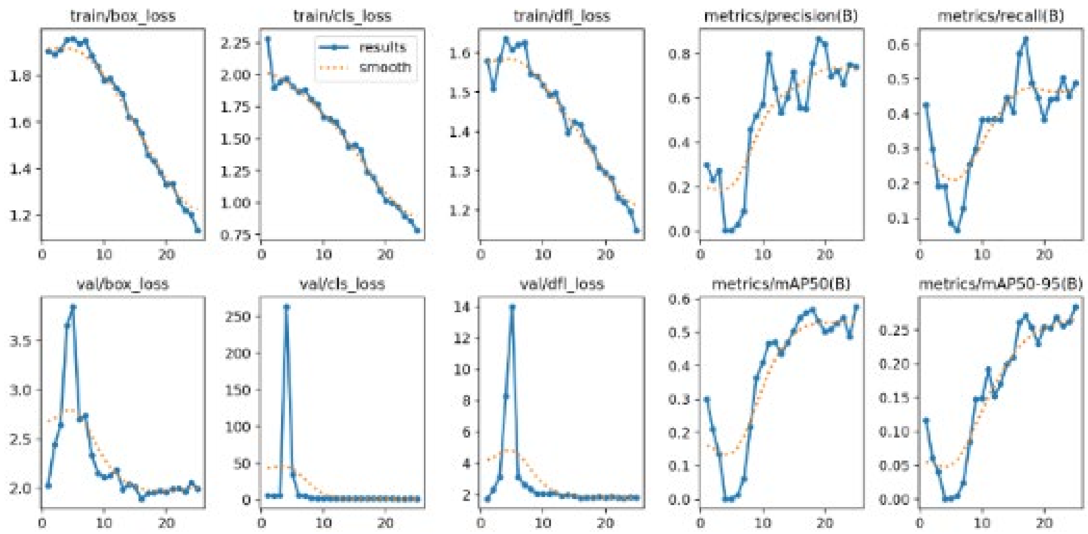
```
TensorBoard: Start with 'tensorboard --logdir runs/detect/train', view at http://localhost:6006/
Freezing layer 'model.22.dfl.conv.weight'
AMP: running Automatic Mixed Precision (AMP) checks with YOLOv8n...
Downloading https://github.com/ultralytics/assets/releases/download/v8.1.0/yolov8n.pt to 'yolov8n.pt'...
100%|██████████| 6.23M/6.23M [00:00<00:00, 89.1MB/s]
AMP: checks passed ✅
train: Scanning /content/gdrive/MyDrive/pan_can/dataset/labels/train/cancer_p.cache... 155 images, 0 backgrounds, 0 corrupt: 100%|██████████| 155/155 [00:00<?, ?it/s]
albumentations: Blur(p=0.01, blur_limit=(3, 7)), MedianBlur(p=0.01, blur_limit=(3, 7)), ToGray(p=0.01), CLAHE(p=0.01, clip_limit=(1, 4.0), tile_grid_size=(8, 8))
val: Scanning /content/gdrive/MyDrive/pan_can/dataset/labels/train/cancer_p.cache... 155 images, 0 backgrounds, 0 corrupt: 100%|██████████| 155/155 [00:00<?, ?it/s]
Plotting labels to runs/detect/train/labels.jpg...
optimizer: 'optimizer=auto' found, ignoring 'lr0=0.01' and 'momentum=0.937' and determining best 'optimizer', 'lr0' and 'momentum' automatically...
optimizer: AdamW(lr=0.002, momentum=0.9) with parameter groups 57 weight(decay=0.0), 64 weight(decay=0.0005), 63 bias(decay=0.0)
TensorBoard: model graph visualization added ✅
Image sizes 640 train, 640 val
Using 2 dataloader workers
Logging results to runs/detect/train
Starting training for 2 epochs...

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
        1/2       2.4G      3.695       5.04      4.182         17        640: 100%|██████████| 10/10 [00:09<00:00,  1.04it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 5/5 [00:03<00:00,  1.49it/s]                   all        155        170   0.000108     0.0294
```

```
Results saved to runs/detect/train3
```

- Listing Training Results

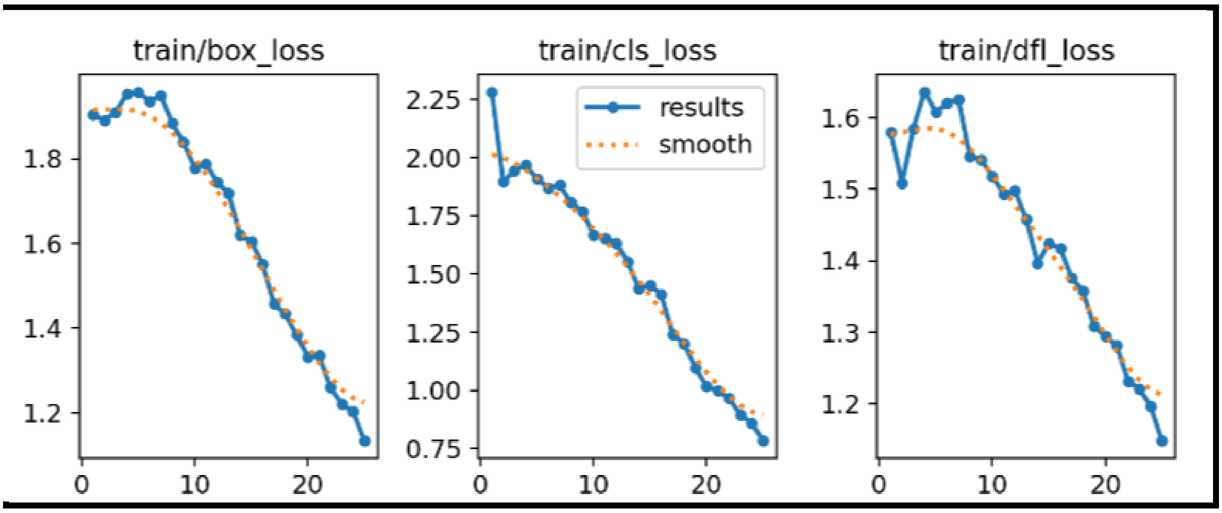
After training the model, we list the contents of the "runs/detect/train6/" directory. This contains various training-related files and artifacts, such as model weights, logs, and metrics.

```
args.yaml                                         P_curve.png        train_batch2.jpg
confusion_matrix_normalized.png                   PR_curve.png       train_batch420.jpg
confusion_matrix.png                              R_curve.png        train_batch421.jpg
events.out.tfevents.1702648999.75cd95999981.6856.0  results.csv      train_batch422.jpg
F1_curve.png                                      results.png        val_batch0_labels.jpg
labels_correlogram.jpg                            train_batch0.jpg   val_batch0_pred.jpg
labels.jpg                                        train_batch1.jpg   weights
```



- Accuracy Metric Curves

- **box_loss:** This curve represents the error in predicting the bounding boxes around detected pancreas. The y-axis shows the magnitude of loss, and x-axis indicates epochs or iterations. A lower value indicates that the model is getting better at accurately drawing boxes around the desired.
- **cls_loss:** This curve illustrates classification loss, indicating how well the model is classifying objects within those predicted bounding boxes as cancer or not. A decrease in this value over epochs (x-axis) means improved accuracy in classification.
- **dfl_loss:** This curve represents the focal loss, which is a modification of the standard cross-entropy loss that is used to address class imbalance in object detection tasks. The y-axis shows the magnitude of loss, and x-axis indicates epochs or iterations. A lower value indicates that the model is getting better at detecting cancer.



- **Precision Curve (Top Left):** This graph shows how precise the model is, meaning how many of the detected objects are actual pancreas . A higher precision indicates fewer false positives.

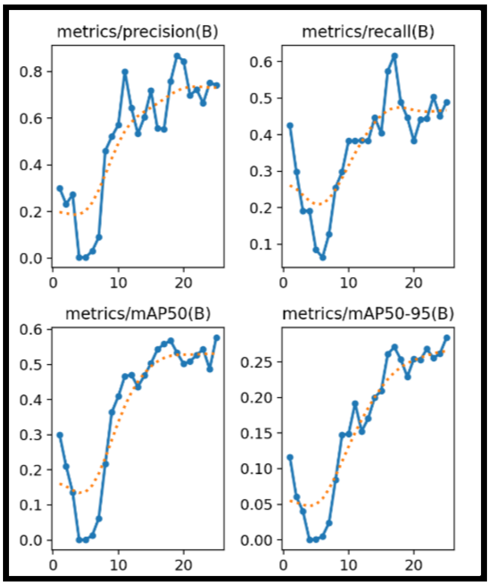
Precision = TP / (TP + FP)

- **Recall Curve (Top Right):** This curve represents the recall rate, indicating how many actual pancreas in the images were detected by the model. A higher recall means that the model is good at identifying pancreas but may also include some false positives.

Recall = TP / (TP + FN)

- **mAP50 Curve (Bottom Left):** mAP50 stands for mean Average Precision at 50% IoU (Intersection over Union). It gives us an idea about the accuracy of our model; higher values indicate better performance in terms of both precision and recall but only considering detections as true positives if they have an IoU above 50% with ground truths.

- **mAP50-95 Curve (Bottom Right):** This is a metric which takes into account detections as true positives only if they have an IoU between 50% and 95% with ground truths. It provides a broader perspective on the model's performance across different levels of detection strictness.



- Programmatic Predictions

In the final step, we load the YOLO model using the YOLO class and make predictions on a specific image programmatically.

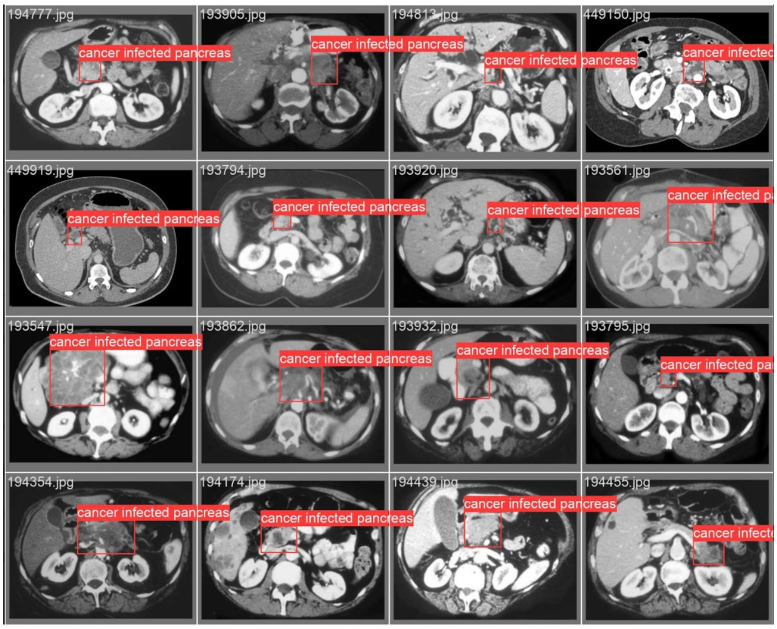
```
Speed: 1.5ms preprocess, 163.1ms inference, 6.1ms postprocess per image at shape (1, 3, 224, 224)
Results saved to runs/detect/predict2
1 label saved to runs/detect/predict2/labels
[ultralytics.engine.results.Results object with attributes:

 boxes: ultralytics.engine.results.Boxes object
 keypoints: None
 masks: None
```

```
orig_img: array([[[255, 255, 255],
        [255, 255, 255],
        [255, 255, 255],
        ...,
        [255, 255, 255],
        [255, 255, 255],
        [255, 255, 255]],

       [[255, 255, 255],
        [255, 255, 255],
        [255, 255, 255],
        ...,
        [255, 255, 255],
        [255, 255, 255],
        [255, 255, 255]],
```

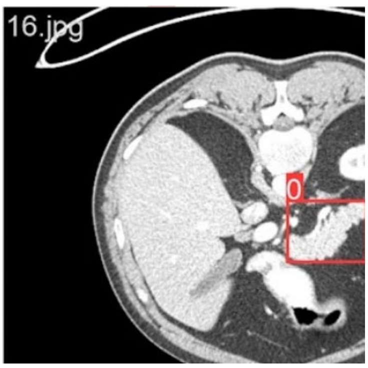
```
       [[202, 208, 213],
        [202, 208, 213],
        [202, 208, 213],
        ...,
        [223, 231, 231],
        [223, 231, 231],
        [223, 231, 231]],

       [[202, 208, 213],
        [202, 208, 213],
        [202, 208, 213],
        ...,
        [223, 231, 231],
        [223, 231, 231],
        [223, 231, 231]],

       [[202, 208, 213],
        [202, 208, 213],
        [202, 208, 213],
        ...,
        [223, 231, 231],
        [223, 231, 231],
        [223, 231, 231]]], dtype=uint8)
orig_shape: (460, 460)
```

```
probs: None
save_dir: 'runs/detect/predict2'
speed: {'preprocess': 1.5411376953125, 'inference': 163.100004196167, 'postprocess': 6.129026412963867}]
```

PANCREATIC ADENOCARCINOMa



Healthy pancreas

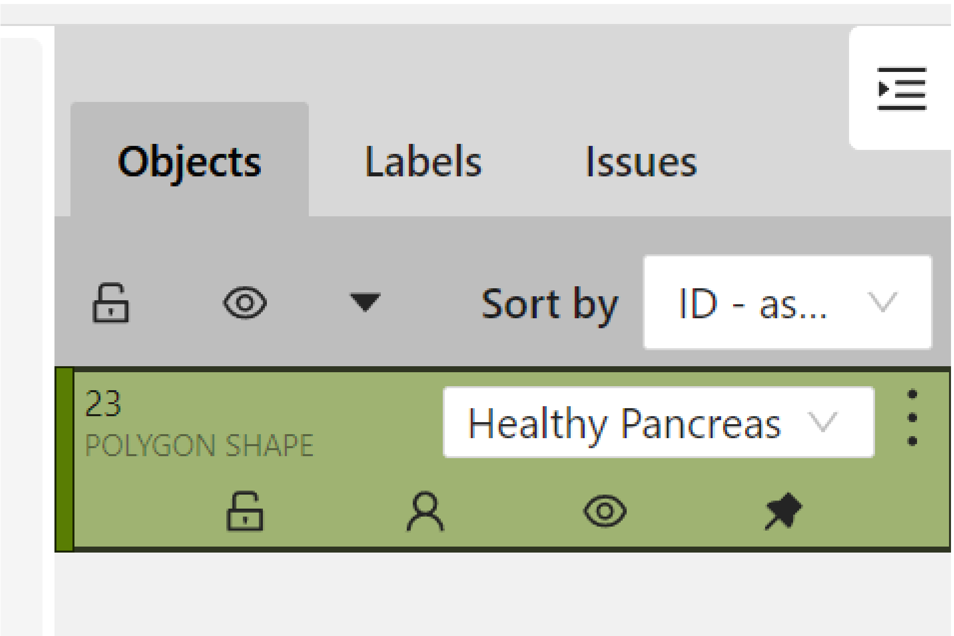


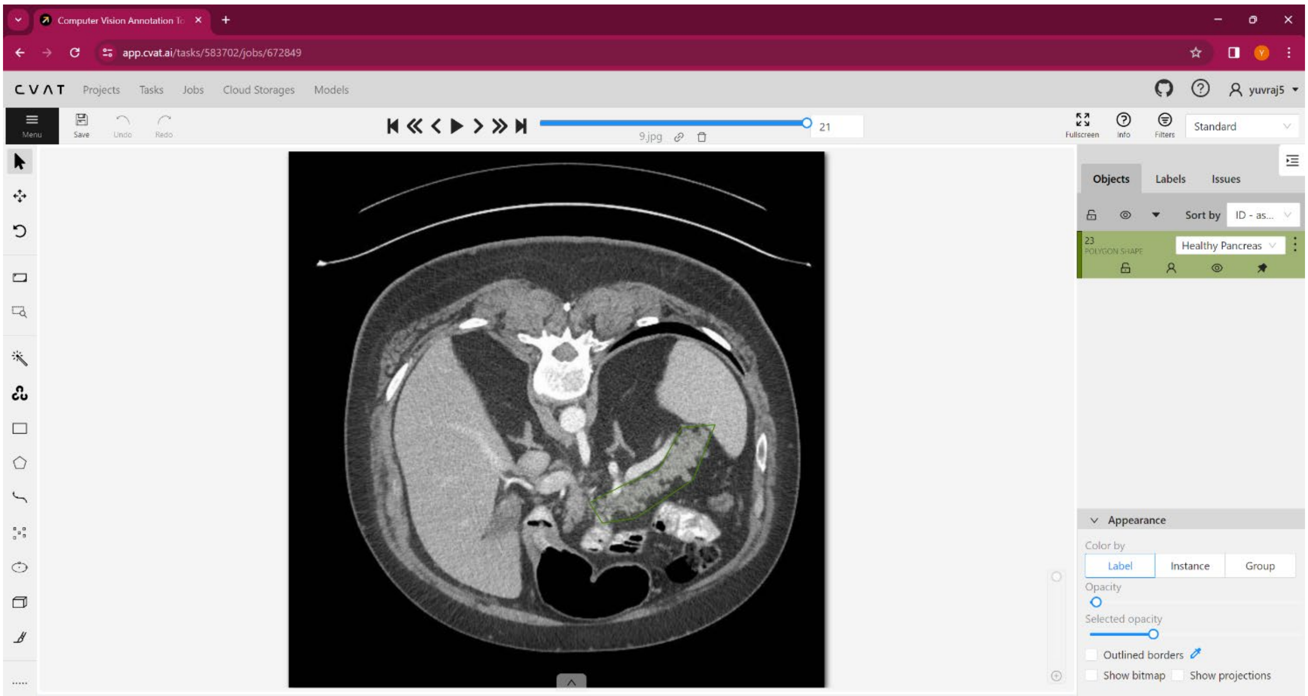
DATA USED – (Private and confidential)

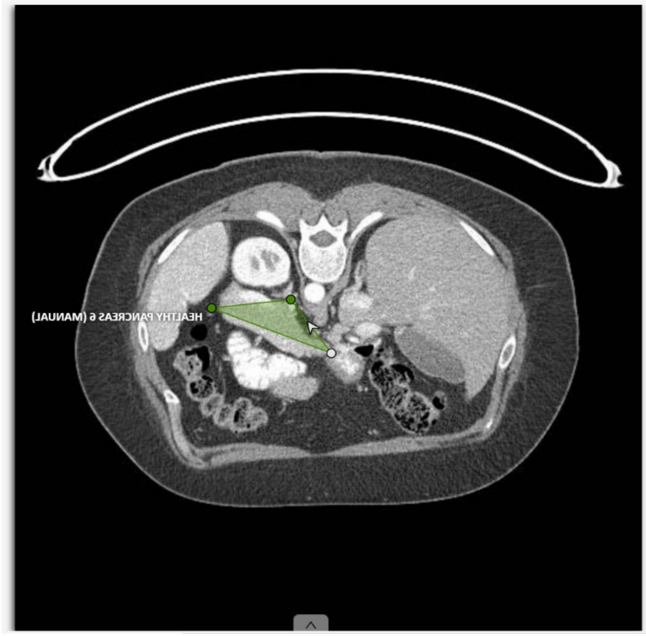
- 133 CT images of cancer patients
- 22 healthy patients

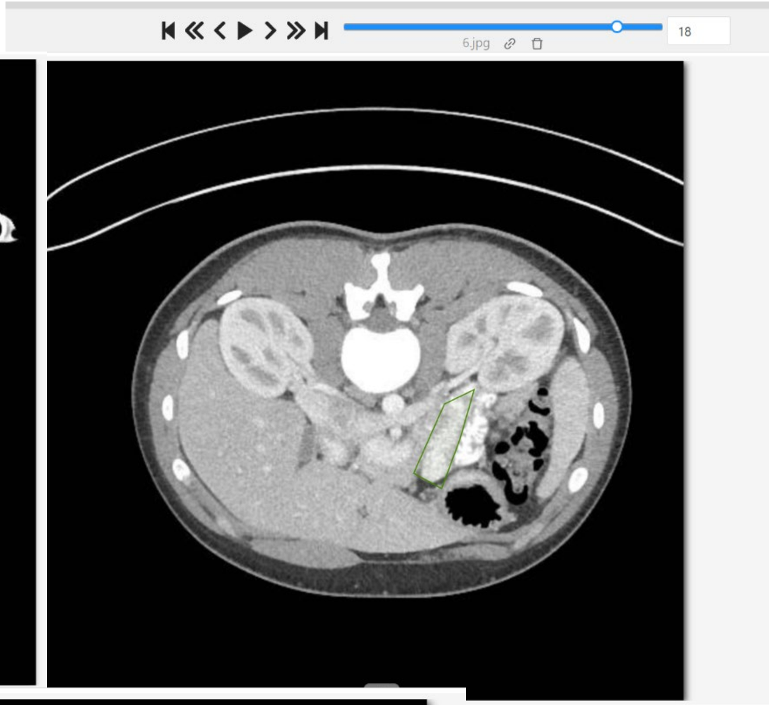
Annotations of data

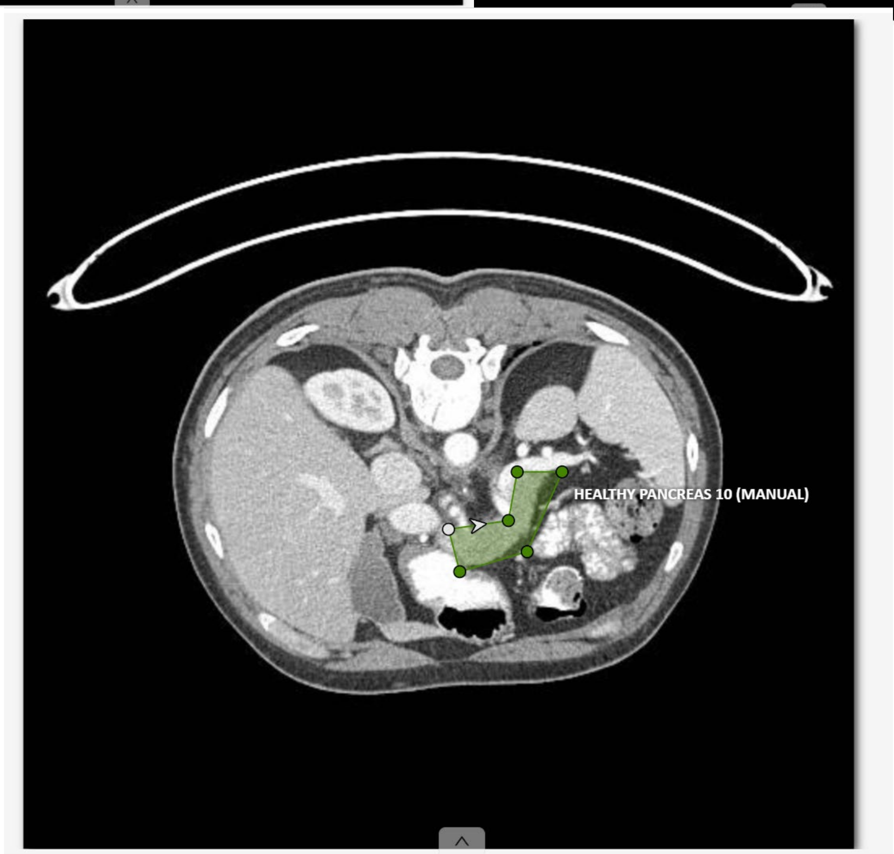
HEALTHY PANCREAS



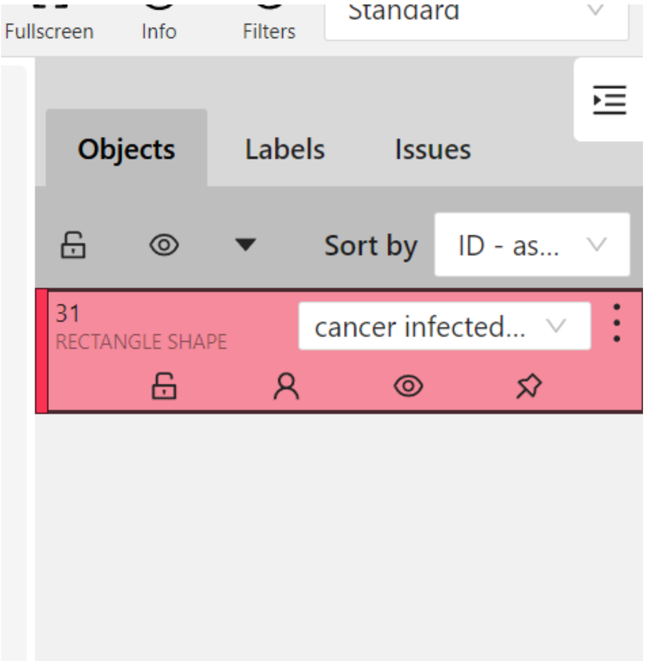


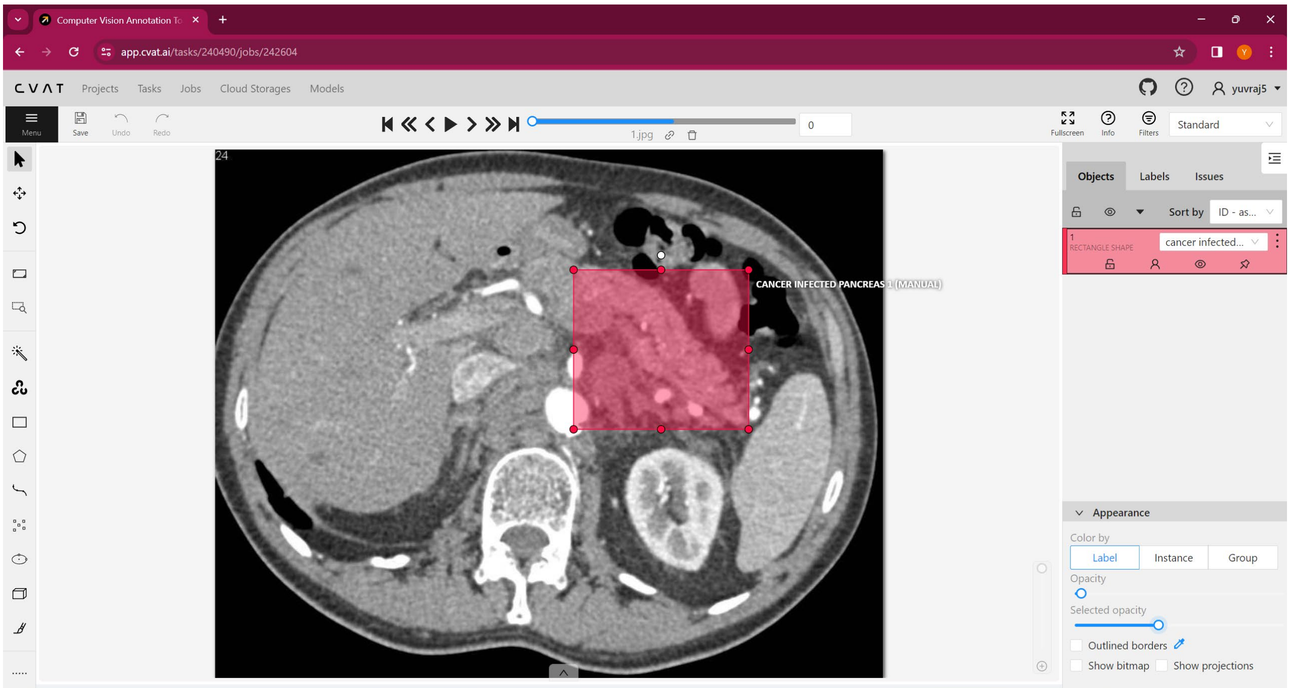

HEALTHY PANCREAS 6 (MANUAL)

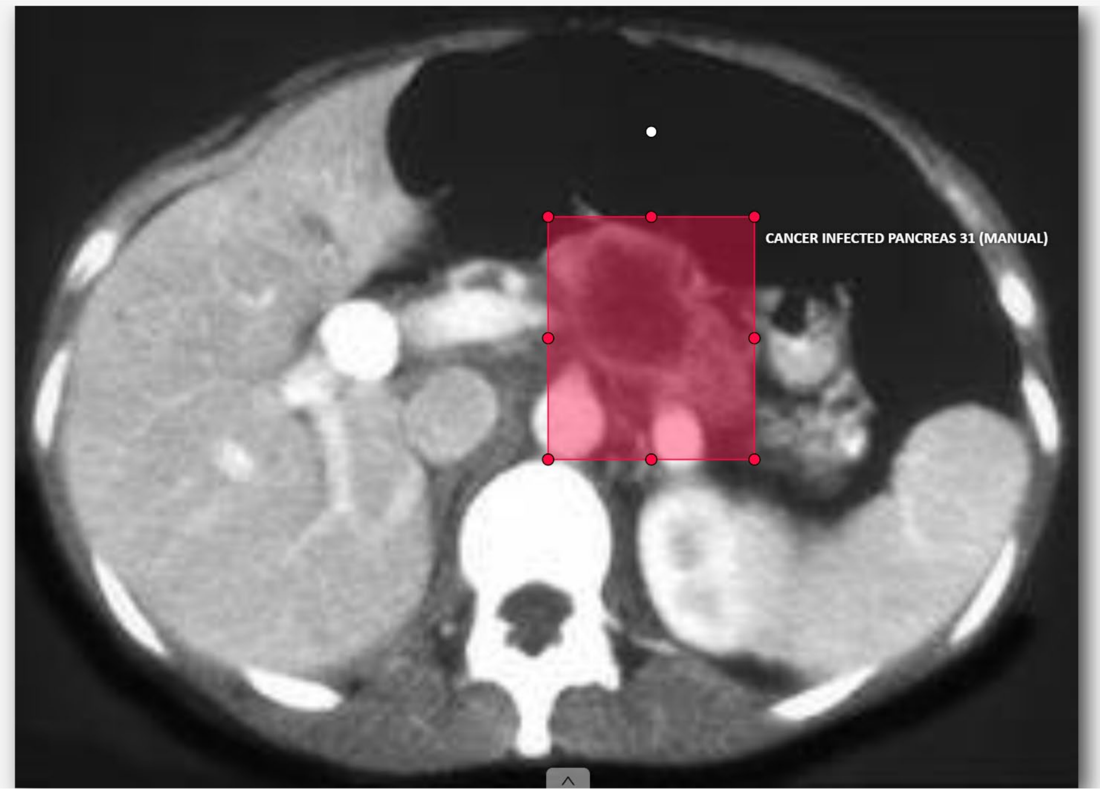

18
6.jpg


HEALTHY PANCREAS 10 (MANUAL)

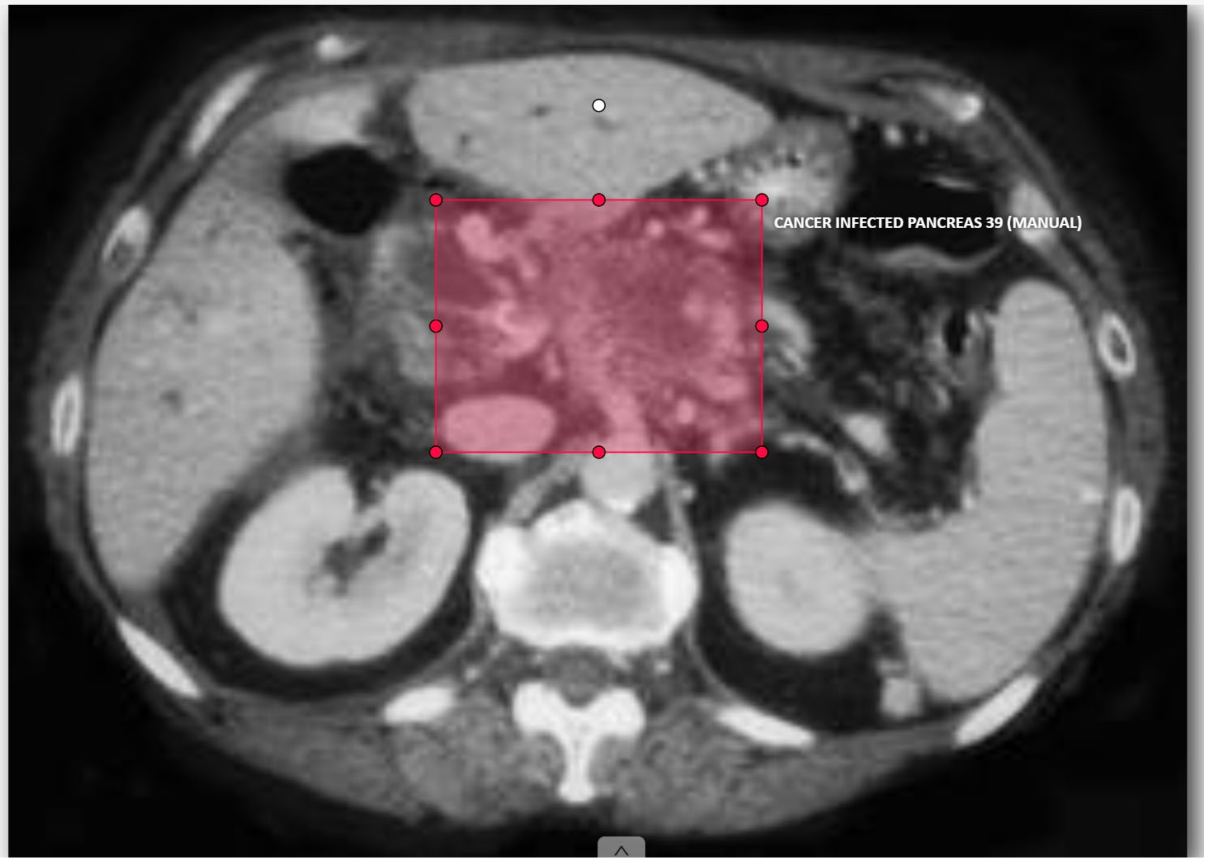
CANCER – Pancreatic adenocarcinoma

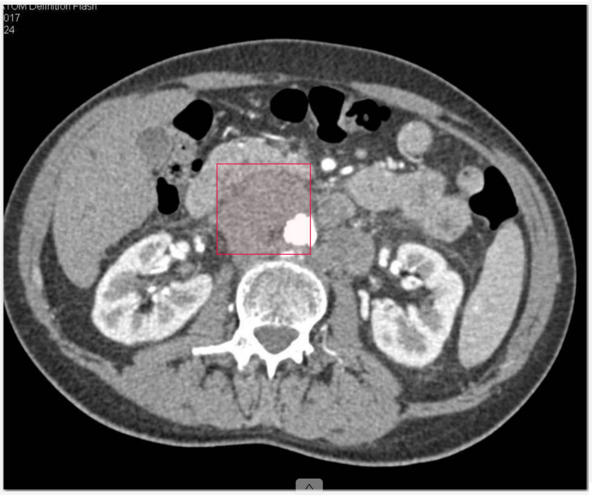


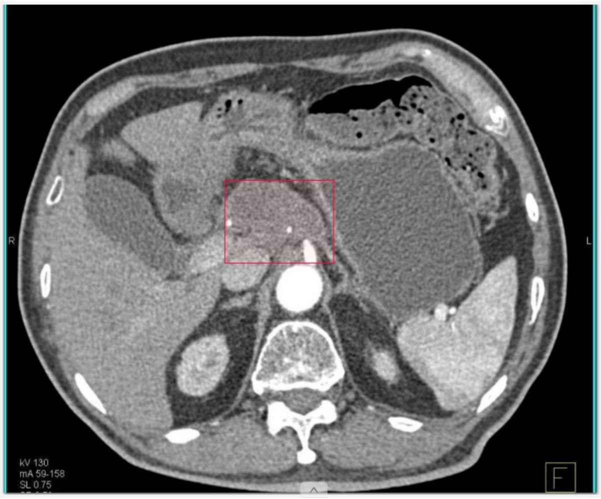



CANCER INFECTED PANCREAS 31 (MANUAL)

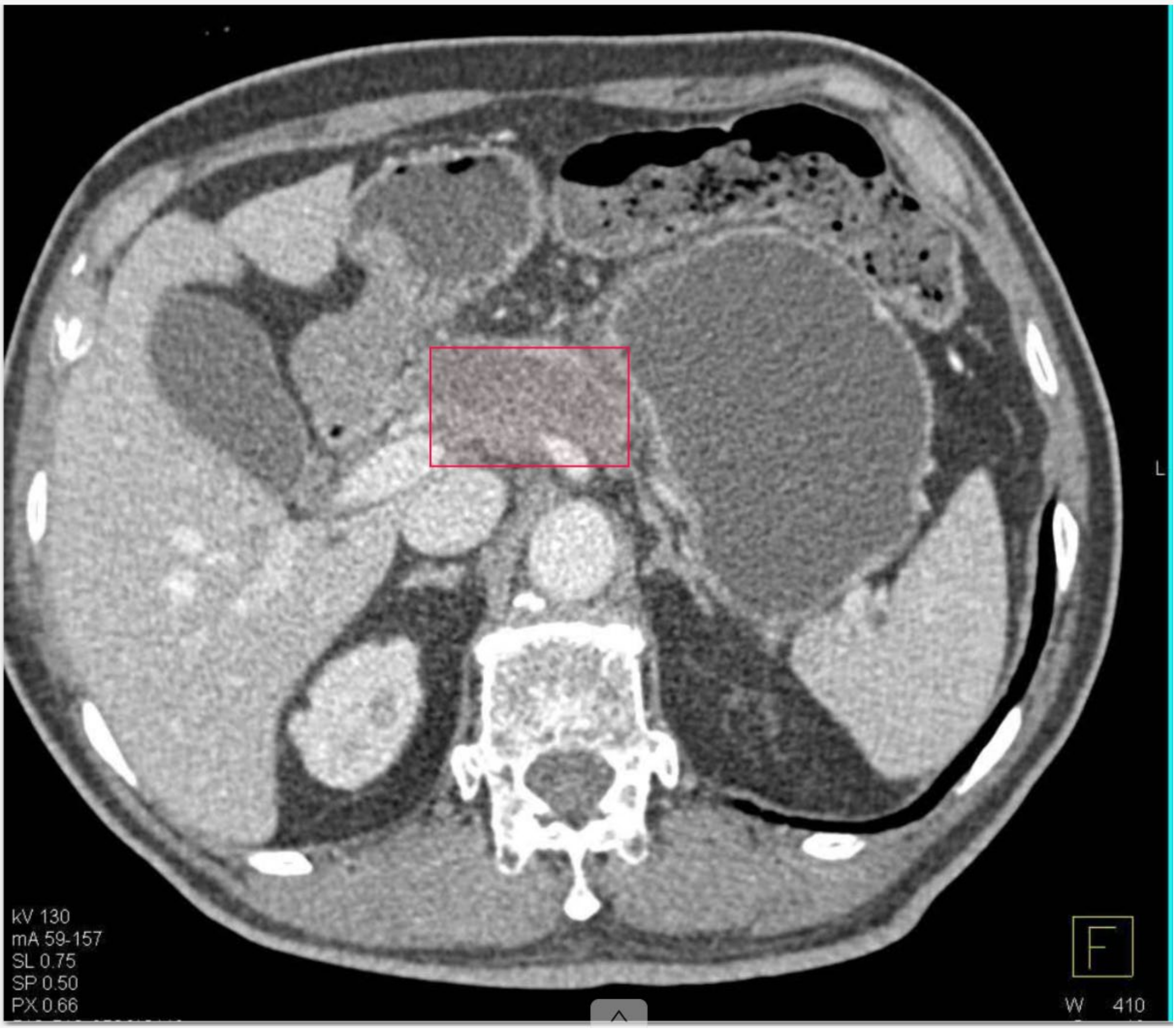

CANCER INFECTED PANCREAS 23 (MANUAL)


CANCER INFECTED PANCREAS 39 (MANUAL)





R
L
kV 130
mA 59-158
SL 0.75
F



L
kV 130
mA 59-157
SL 0.75
SP 0.50
PX 0.66
F
W 410